

```
100100100011101110101001001011
010011110001010011101000111001
101111101001110010010011001001
000111011101010010010110100111
100010100111010001110011011111
0101101010010010110100111 001
01001110100011100110111110 1
110010010010010010001 011
100100101101001111000
10001110011011111 1 0 1 0 0 0 1 0 0 1
0010010011 011 0 1 0 1 0 0 1 0 1
```

Manual Pages

TABLE OF CONTENTS**Preface**

| | |
|---------------------|----|
| edrcintro (1) | 12 |
|---------------------|----|

A

| | |
|------------------------------|----|
| ansi2txt (3) | 59 |
| appendpdf (3) | 60 |
| apply2file (1) | 61 |
| apply2sw_inventory (1) | 64 |
| approot (3) | 69 |
| apprevison (3) | 70 |
| apropos (1) | 72 |
| ascii (3) | 74 |
| asup (1) | 77 |
| awk (1) | 80 |

B

| | |
|------------------------|-----|
| banner (3) | 84 |
| bash (1) | 86 |
| batteryalert (1) | 150 |
| bc (1) | 153 |
| bget (3) | 164 |
| binprobe (1m) | 172 |
| binprobe.dat (4) | 174 |
| bunzip2 (1) | 176 |
| busy (1) | 181 |
| bwcreate (1m) | 183 |
| bwcreate.cfg (4) | 189 |
| bzcat (1) | 193 |
| bzdiff (1) | 198 |
| bzgrep (1) | 199 |
| bzip2 (1) | 200 |
| bzip2recover (1) | 205 |
| bzmore (1) | 210 |

C

| | |
|-------------------------|-----|
| catcomp (1) | 212 |
| ccrypt (1) | 214 |
| cfg2html (1m) | 221 |
| checkopt (3) | 225 |
| checkopt.h (3) | 229 |
| choice (3) | 233 |
| cltrash (1) | 236 |
| cmdlist (1m) | 240 |
| cmmon (1m) | 246 |
| cmswitch (1m) | 248 |
| comm (1) | 250 |
| compatibility (1) | 251 |
| connect (3) | 259 |
| consolidate (1) | 261 |
| contrib (1m) | 263 |
| contrib.doc (1m) | 268 |
| contrib.edrc (1m) | 276 |
| cpanm (1) | 289 |
| cpio (1) | 296 |
| crfile (1) | 300 |

| | |
|-----------------------|-----|
| crond (3) | 303 |
| cronhandler (1m) | 305 |
| cronhandler.cfg (4) | 313 |
| crontab (4) | 315 |
| crypt (1) | 319 |
| csv (3) | 320 |
| csvcat (3) | 331 |
| csvq (3) | 333 |
| csv2worksheet (3) | 335 |
| D | |
| datalist.dat (4) | 341 |
| daemon_wrapper (1) | 344 |
| days (3) | 347 |
| dbrep (1) | 350 |
| dbrep.cfg (4) | 353 |
| diff (1) | 355 |
| dig (1) | 359 |
| directories (3) | 364 |
| dmidecode (3) | 366 |
| dos2ux (3) | 369 |
| du_index (4) | 371 |
| du_report (1) | 373 |
| duvi (1) | 376 |
| E | |
| ecrontab (1) | 378 |
| ecronnext (1) | 380 |
| edrc (1m) | 382 |
| edrc.cfg (4) | 399 |
| edrc.help (4) | 407 |
| edrcinit (1m) | 408 |
| edrcinit.cfg (4) | 412 |
| edrcinit.handler (3) | 417 |
| edrcenv (1) | 419 |
| edrclicense (4) | 421 |
| edrcman (1) | 429 |
| edrcman.cfg (4) | 433 |
| edrcman.map (4) | 435 |
| edrcpack (1m) | 437 |
| edrcports (4) | 439 |
| edrcperm.no_shell (3) | 444 |
| edrcrevision.cfg (4) | 446 |
| edrcroot (3) | 448 |
| edrcscripts (1m) | 449 |
| edrcsetup (1m) | 454 |
| env (4) | 471 |
| envpasswdstrip (1) | 475 |
| epub2pdf (1) | 477 |
| eshell (1) | 479 |
| eterm (1) | 481 |
| expect (3) | 484 |
| exiftool (1) | 509 |
| exrc (4) | 544 |

F

| | |
|--------------------|-----|
| fcreate (3) | 546 |
| fields2swvi (3) | 548 |
| filedist (1) | 551 |
| filedist.cfg (4) | 556 |
| filedist.block (4) | 559 |
| filegrep (1) | 563 |
| filelink (1) | 565 |
| filesize (3) | 567 |
| filewatch (3) | 569 |
| fit2width (3) | 571 |
| fnmatch (3) | 574 |
| fmatch (3) | 577 |
| freespace (3) | 580 |
| fssum (1) | 582 |
| fssum.cfg (4) | 584 |
| ftps (1) | 587 |

G

| | |
|-------------------|-----|
| gawk (1) | 590 |
| gecos (3) | 612 |
| getfilesystem (3) | 614 |
| gethostbyname (1) | 616 |
| getmountpoint (3) | 618 |
| gid (3) | 620 |
| glibc.version (3) | 622 |
| grep (1) | 624 |
| group (3) | 633 |
| gs (3) | 635 |
| gzip (1) | 641 |

H

| | |
|---------------------|-----|
| h2 (1) | 646 |
| histlist (3) | 649 |
| homedir (3) | 651 |
| hostaliases (3) | 653 |
| hostlist (3) | 655 |
| hostlist.cfg (4) | 660 |
| hostlist.dat (4) | 666 |
| hostlistdat2cfg (3) | 669 |
| html2mht (3) | 678 |
| hwinventory (1) | 680 |

I

| | |
|----------------------|-----|
| ident (3) | 682 |
| indent (3) | 686 |
| info (1) | 690 |
| ini (1) | 692 |
| ini.cfg (4) | 696 |
| input (3) | 698 |
| input_targets (3) | 703 |
| ipcalc (1) | 705 |
| ipsort (1) | 708 |
| is_config_byhand (3) | 710 |
| is_existing (3) | 713 |
| is_osid (3) | 715 |

| | |
|-------------------------|-----|
| is_permitted (3) | 717 |
| is_running (3) | 721 |
| is_up (3) | 723 |
| is_user (3) | 725 |
| is_weekend (3) | 727 |
| is_writeable (3) | 729 |
| J | |
| java (1) | 731 |
| java_wrapper (1) | 733 |
| java_wrapper.cfg (4) | 735 |
| job (1) | 737 |
| job.cfg (4) | 743 |
| joblog (1) | 745 |
| jobstart (1) | 748 |
| jobwatch (1) | 756 |
| jq (3) | 759 |
| K | |
| kalc (1) | 801 |
| ksh (1) | 802 |
| kshell (3) | 839 |
| kshrc (4) | 842 |
| ksh_wrapper (1) | 844 |
| ksh_wrapper.cfg (4) | 847 |
| L | |
| leo (1) | 849 |
| libexpect (3) | 853 |
| lbanner (3) | 861 |
| lgcheckd (1m) | 863 |
| lgcheckd.cfg (4) | 872 |
| lgcheckd.state.db (4) | 877 |
| lgcpattern (3) | 880 |
| listtemp (3) | 884 |
| ll (1) | 887 |
| llcomp (1) | 889 |
| locate (1) | 891 |
| locatedb (4) | 894 |
| locations (1) | 896 |
| locations.cfg (4) | 899 |
| log (3) | 902 |
| logcat (1) | 905 |
| logcheckd (1m) | 911 |
| logcheckd.cfg (4) | 923 |
| logcheckd.interface (3) | 931 |
| logcheckd.pattern (4) | 935 |
| logcheckd.style (4) | 938 |
| logcut (1) | 942 |
| logcut.cfg (4) | 944 |
| loggrep (1) | 946 |
| logrotate (1) | 952 |
| logrotate.cfg (4) | 954 |
| logtail (1) | 956 |
| logview (1) | 960 |
| log_files.cfg (4) | 964 |

| | |
|---------------------|------|
| lots (1m) | 966 |
| lots.cfg (4) | 979 |
| lotsctl (1) | 982 |
| lotsctl.cfg (4) | 985 |
| lscomp (1) | 988 |
| lscp (1) | 990 |
| lsmv (1) | 992 |
| lsof (1) | 994 |
| lspm (1) | 1029 |
| lua (1) | 1031 |
| luac (1) | 1033 |
| luaenv (3) | 1034 |
| luarocks (1) | 1040 |
| luarocks.cfg (4) | 1041 |
| luaversion (3) | 1044 |
| lua_wrapper (1) | 1046 |
| lua_wrapper.cfg (4) | 1049 |
| lynx (1) | 1051 |
| M | |
| mail_file (1) | 1067 |
| mail_file.cfg (4) | 1070 |
| maketemp (3) | 1073 |
| makeuser (1) | 1076 |
| makeuser.cfg (4) | 1083 |
| man2html (3) | 1085 |
| mandir (4) | 1087 |
| manpages (4) | 1091 |
| manvi (1) | 1108 |
| manvi.cfg (4) | 1110 |
| mc (1) | 1112 |
| mcedit (1) | 1155 |
| mcview (1) | 1163 |
| md2html (3) | 1165 |
| md5string (3) | 1167 |
| mediawiki (3) | 1169 |
| mkuser (3) | 1172 |
| msg (3) | 1175 |
| msmtp (3) | 1178 |
| N | |
| name (1) | 1187 |
| nano (1) | 1191 |
| nanorc (4) | 1194 |
| nc (3) | 1198 |
| nginx (3) | 1201 |
| nmap (1) | 1204 |
| nologin (1) | 1243 |
| nping (1) | 1245 |
| O | |
| omniutil (1m) | 1268 |
| omniutil.cfg (4) | 1273 |
| os_wrapper (1) | 1275 |
| osid (3) | 1278 |
| osid.cfg (4) | 1282 |

| | |
|----------------------|------|
| osid.dat (4) | 1285 |
| osid.probe (3) | 1288 |
| osup (1) | 1290 |
| outex (1) | 1293 |

P

| | |
|-----------------------------|------|
| pack (1m) | 1296 |
| pack.cfg (4) | 1305 |
| passwdcombine (1) | 1311 |
| passwdsort (1) | 1314 |
| passwdsyncd (1m) | 1316 |
| passwdsyncd.cfg (4) | 1320 |
| passwdsyncd_apply (3) | 1324 |
| patchinstall (1m) | 1327 |
| patchinstall.cfg (4) | 1332 |
| pdfmetaedit (1) | 1335 |
| pdfscissors (1) | 1337 |
| perlenv (3) | 1339 |
| perlpod (4) | 1347 |
| perl_modules (3) | 1355 |
| perlversion (3) | 1361 |
| perl_wrapper (1) | 1363 |
| perl_wrapper.cfg (4) | 1366 |
| pf_wrapper (1) | 1368 |
| pip (1) | 1373 |
| pip3 (1) | 1375 |
| pip3-check (1) | 1378 |
| pip3-download (1) | 1381 |
| pip3-freeze (1) | 1384 |
| pip3-hash (1) | 1385 |
| pip3-help (1) | 1386 |
| pip3-install (1) | 1389 |
| pip3-list (1) | 1394 |
| pip3-search (1) | 1396 |
| pip3-show (1) | 1397 |
| pip3-uninstall (1) | 1398 |
| pip3-wheel (1) | 1399 |
| pkgdir (1m) | 1402 |
| pkg_hostname (3) | 1416 |
| pkg_hostname.cfg (4) | 1420 |
| pkzip (1) | 1423 |
| pid (1) | 1430 |
| pl (1) | 1432 |
| pmdesc (1) | 1441 |
| pod2html (3) | 1443 |
| portscan (3) | 1446 |
| print_header (3) | 1448 |
| print_index (3) | 1451 |
| print_list (3) | 1454 |
| program.h (3) | 1457 |
| pscount (3) | 1460 |
| pscount.kp (4) | 1465 |
| psjoin (3) | 1466 |
| pslist (3) | 1468 |
| ps2pdf (3) | 1472 |

| | |
|----------------------------|------|
| pstree (1) | 1474 |
| psup (1) | 1477 |
| purgetemp (3) | 1480 |
| pwcrypt (1) | 1483 |
| pwsafe (3) | 1485 |
| pythonenv (3) | 1488 |
| pythonversion (3) | 1494 |
| python_wrapper (1) | 1496 |
| python_wrapper.cfg (4) | 1499 |
| R | |
| random (3) | 1501 |
| rcat (1) | 1503 |
| rcmd (1) | 1505 |
| rcmd.cfg (4) | 1508 |
| rcomm (1) | 1511 |
| rdiff (1) | 1514 |
| readline (3) | 1517 |
| regexintro (4) | 1519 |
| rel2abs (3) | 1524 |
| remote_copy (3) | 1526 |
| remote_shell (3) | 1529 |
| removetemp (3) | 1532 |
| repeat (3) | 1535 |
| resolve_targetlist (3) | 1537 |
| revision (1) | 1539 |
| rl (1) | 1542 |
| rnano (1) | 1544 |
| role_option (1) | 1545 |
| role_option.cfg (4) | 1548 |
| rosid (3) | 1554 |
| rsat (1) | 1557 |
| rsat.cfg (4) | 1559 |
| rssh (1) | 1561 |
| rsync (1) | 1564 |
| rsyncd.conf (4) | 1606 |
| S | |
| sat (1) | 1617 |
| sav (1) | 1620 |
| schedule.dat (4) | 1623 |
| screen (1) | 1626 |
| scriptextract (3) | 1673 |
| scriptgrep (1) | 1675 |
| scripthheadersync (1) | 1677 |
| scriptmenupath (3) | 1679 |
| scriptrevision (3) | 1681 |
| scriptsequence (3) | 1683 |
| scripttitle (3) | 1686 |
| seconds (3) | 1688 |
| sectioncat (3) | 1691 |
| sedlline (1) | 1693 |
| select_columns (3) | 1703 |
| server_environment (3) | 1706 |
| server_environment.cfg (4) | 1710 |
| sh (1) | 1714 |

| | |
|---------------------|------|
| shell (1) | 1724 |
| shell.cfg (4) | 1731 |
| shellinaboxd (3) | 1735 |
| shlib (3) | 1741 |
| sortc (3) | 1743 |
| sortv (3) | 1745 |
| sparse (3) | 1747 |
| splitvt (1) | 1749 |
| sqlite (1) | 1752 |
| ssh-exec (1) | 1755 |
| ssh-exec.cfg (4) | 1761 |
| ssh-keyadd (1m) | 1764 |
| ssh-keyadd.cfg (4) | 1767 |
| ssh-keyadd.pub (4) | 1770 |
| sshpas (1) | 1772 |
| stat (1) | 1774 |
| streamcat (3) | 1778 |
| strings.h (3) | 1780 |
| strace (1) | 1782 |
| svi (1) | 1787 |
| sw_inventory (4) | 1789 |
| sw_report (1) | 1792 |
| swvi (1) | 1795 |
| symlink (1) | 1799 |
| sys (1) | 1802 |
| sysconfig (1) | 1805 |
| sysconfig.cfg (4) | 1825 |
| sysconfig.tools (3) | 1827 |
| sysinfo (1) | 1831 |
| syscp (1) | 1835 |
| sysdiff (1) | 1838 |
| syspoll (1) | 1840 |
| syspoll.cfg (4) | 1857 |
| sysvi (1) | 1860 |
| sys2html (1m) | 1862 |
| sys2html.dat (4) | 1866 |
| sys2html.inf (4) | 1868 |

T

| | |
|------------------|------|
| tcpdump (1) | 1870 |
| termcap (4) | 1889 |
| textblock (3) | 1897 |
| textcolor (3) | 1899 |
| thttpd (3) | 1901 |
| timer (1) | 1907 |
| timezone (3) | 1910 |
| timezone.cfg (4) | 1913 |
| timezone.dat (4) | 1915 |
| title (1) | 1917 |
| today (3) | 1919 |
| tolower (3) | 1921 |
| tomorrow (3) | 1923 |
| tpl (1) | 1925 |
| tpl.cfg (4) | 1928 |
| top (1) | 1930 |

| | |
|---------------------------|------|
| toupper (3) | 1934 |
| tput_examples (3) | 1936 |
| traceroute (1m) | 1939 |
| trash (1) | 1942 |
| trash.cfg (4) | 1946 |
| truss (1) | 1948 |
| tscat (3) | 1954 |
| tsize (3) | 1957 |
| ttyplay (1) | 1959 |
| ttyrec (1) | 1960 |
| ttytime (1) | 1961 |
| tty_columns (3) | 1962 |
| tty_variable (3) | 1964 |
| tzdump (3) | 1966 |
| U | |
| uid (3) | 1968 |
| uncbz (1) | 1970 |
| uncbz2 (1) | 1972 |
| uncgz (1) | 1974 |
| unzip (1) | 1976 |
| undeb (1) | 1978 |
| uniqupath (3) | 1980 |
| unrpm (1) | 1983 |
| untbz (1) | 1985 |
| untbz2 (1) | 1987 |
| untgz (1) | 1989 |
| untrash (1) | 1991 |
| untxz (1) | 1993 |
| unzip (1) | 1995 |
| updatedb (1m) | 2007 |
| usage (1) | 2009 |
| usage.list (4) | 2012 |
| user (3) | 2014 |
| userclass.index (4) | 2016 |
| user_info (1) | 2018 |
| user_info.cfg (4) | 2021 |
| utility.h (3) | 2026 |
| uudecode (1) | 2028 |
| uuencode (1) | 2031 |
| uuencode (4) | 2033 |
| ux2dos (3) | 2034 |
| V | |
| vadsp (1) | 2036 |
| volume.dat (4) | 2038 |
| vsav (1) | 2040 |
| vsdfml (3) | 2043 |
| vvi (1) | 2047 |
| W | |
| wa2lc (3) | 2049 |
| wa2ledrc_edrcapi (1) | 2051 |
| wa2ledrc_report (1) | 2054 |
| wa2ledrc_shellinaboxd (1) | 2060 |
| wa2l_util (3) | 2062 |

| | |
|------------------|------|
| watchdog (1) | 2065 |
| watchdog.cfg (4) | 2070 |
| wget (1) | 2073 |
| whatis (1) | 2095 |
| whereami (1) | 2096 |
| whoisin (1) | 2099 |
| whoisin.list (4) | 2101 |
| winexe (1) | 2103 |
| wmic (1) | 2105 |
| woist (1) | 2107 |
| X | |
| xbdf (1m) | 2109 |
| xlog (1) | 2111 |
| xml2csv (3) | 2113 |
| xml2json (3) | 2116 |
| xmore (1) | 2118 |
| xpid (1) | 2120 |
| xtee (1) | 2122 |
| xz (1) | 2124 |
| Y | |
| yesterday (3) | 2143 |
| ypxfr_all (1) | 2145 |
| Z | |
| zcat (1) | 2147 |
| zdiff (1) | 2152 |
| zgrep (1) | 2153 |
| zip (1) | 2154 |
| zless (1) | 2181 |
| zmore (1) | 2182 |
| zstd (1) | 2184 |
| Appendix | |
| edrcrevision (1) | 2192 |

SEE ALSO

edrc(1), **edrdcintro(1)**

AUTHOR

WA2L/edrc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcintro – Introduction and overview for EDRC, the Enterprise Disaster Recovery Console

AVAILABILITY

WA2L/edrc

DESCRIPTION

EDRC is a package that enables you to recover your environment in an efficient and reliable way. The EDRC is an integral part of the "Emergency Handbook" and a whole disaster recovery plan and process.

This manpage gives an overview over all commands and directories that are part of the WA2L/edrc package.

The command for the **Enterprise Disaster Recovery Console**, the interactive program to control a disaster recovery or also useful to simplify system administration tasks, is **edrc**. See **edrc(1m)** for a detail description of the console.

REVISION-HISTORY

The revision history of WA2L/edrc is in a separate man page. See **edrcrevision(1)** to check the changes between the WA2L/edrc releases.

COMMANDS**A**

ansi2txt(3) converts a stream of vt100 (ansi) codes on the **stdin** into readable text on the **stdout**. A file can be specified instead of **stdin**.

appendpdf(3) Copy the contents of *file2.pdf* to the end of *file1.pdf*. The concatenated file is written to the specified output file or to **stdout**.

apply2file(1) apply a data stream provided via pipe to a target file if the content differs from the target file. This is useful to edit files or to generate files which are system dependent.

Example:

```
banner `hostname` | \
  apply2file -o -u root -g root -p 640 \
  -f /etc/issue

cat /etc/rc.config.d/samba | \
  sed -e 's/RUN_SAMBA=.* /RUN_SAMBA=1/g' | \
  apply2file -o -f /etc/rc.config.d/samba
```

apply2sw_inventory(1)

apply a data stream to a software inventory file. See also **swvi(1)**.

aproot(3)

print installation root of WA2L/edrc.

aprevision(3)

print application revision of the WA2L/edrc package.

apropos(1)

search the whatis database for strings.

ascii(3)

This is not a command, but a documentation of ASCII character set encoded in octal, decimal, and hexadecimal.

asup(1)

application support configuration short start for **edrc**. See **sat(1)** for more information.

awk(1)

manual page of **awk**, pattern-directed scanning and processing language.

B**banner(3)**

print big banner text as known from the login banners.

bash(1)

manual page of **bash**, GNU Bourne-Again SHell.

batteryalert(1)

monitor battery capacity and alert to all terminals on low charge.

binprobe(1m)

probe (compiled) executables and perl scripts if they are able to start on the current operating system.

bget(1)

basic tool to make HTTP GET requests and monitor the results.

busy(1)

print a rotating slash. This command does nothing productive. It might be useful to keep a line open when for instance a firewall cuts the connection if a TCP timeout is reached.

bwcreate(1m)

create a BarbedWire file that contains file properties and checksum of files on filesystems defined in the **bwcreate.cfg(4)** file. The BarbedWire file is used to track changes on (operating system) files.

| | |
|-------------------------|--|
| bzcat (1) | decompresses .bz2 files to stdout. |
| bzdiff (1) | compare bzip2 compressed files. |
| bzgrep (1) | search possibly bzip2 compressed files for a regular expression |
| bunzip2 (1) | unzip a block-sorting .bz2 file. |
| bzip2 (1) | a block-sorting file compressor to generate .bz2 files. |
| bzip2recover (1) | recovers data from damaged bzip2 files. |
| bzmore (1) | file perusal filter for crt viewing of bzip2 compressed text. |

C

| | |
|----------------------|---|
| catcomp (1) | cat a file out of a .tar.gz , .tgz or .cpio.gz file without decompressing the archive. |
| ccrypt (1) | ccrypt is a utility for encrypting and decrypting files and streams. It was designed to replace the standard unix crypt utility, which is notorious for using a very weak encryption algorithm. ccrypt is based on the Rijndael block cipher, which was also chosen by the U.S. government as the Advanced Encryption Standard (AES, see http://www.nist.gov/aes/). This cipher is believed to provide very strong cryptographic security. |
| cfg2html (1m) | This command creates a HTML (and ASCII) system documentation for HP-UX 10+11, AIX, SCO-UX, Solaris and Linux Systems. Plugins for SAP, Oracle, Informix, MC/SG, FibreChannel, TIP/ix, Mass Storage like XP48/256/512, Network Node Manager and OmniBack etc. are included. See also https://cfg2html.com/ . |
| checkopt (3) | check list of used options matches against a table of allowed option combinations. This command is intended to be used to efficiently parse arguments in a script. |
| choice (3) | an interactive choice command for use in own recovery scripts and contributed commands. |
| cltrash (1) | perform cleanup tasks for files trashed with the trash command. |

| | |
|--------------------------|--|
| cmdlist (1m) | list all available commands in WA2L/edrc with some credentials, as: TYPE, location, operating system and start mode. See also compatibility (1). |
| cmmon (1m) | monitor of all MC/SG packages on HP-UX systems having MC/ServiceGuard installed. If an interval of 0 is specified a one time output is provided, else the output is continuously. |
| cmswitch (1m) | switch a MC/SG package to a node. Using this interactive command is much faster then entering all commands needed for a cluster package switch from one to an other node. You can restart a package if you specify the host where the package is currently running as the target node. Be aware that the cmswitch command does not check if it is possible to switch the package to the specified target node, therefore you should check the possible package switch positions first using the cmmon command. |
| compatibility (1) | This is not a command, but a documentation of the "WA2L/edrc command compatibility overview (compatibility matrix)" by Christian Walther. |
| connect (3) | connect is a proxy tool to enable OpenSSH and other TCP/IP utilities to run through SOCKS4/5 and HTTP proxy gateways. The arguments host and port are for the target hostname and port number to connect to. |
| consolidate (1) | filter to consolidate structured unix file content. |
| cpanm (1) | get, unpack build and install modules from CPAN. The cpanm command settings are tuned for the WA2L/edrc package and the Perl environment distributed with the package. |
| cpio (1) | copy file archives in and out; duplicate directory trees. |
| crfile (1) | create a dump file with a specific size. |
| crond (3) | <p>daemon to execute scheduled commands. The crond integrated into WA2L/edrc allows to schedule jobs independent of the system.</p> <p>To add/edit crontab(4) entries, use the ecrontab(1) command and to show the next scheduled commands ecronnext(1).</p> <p>The crond(3) service is handled thru edrcinit(1m).</p> |

| | |
|--------------------------|--|
| cronhandler (1) | handle cron and at jobs in a cluster environment. With cronhandler you can transfer crontab entries and even defined at jobs from one cluster node to an other depending on username. at jobs are time-adjusted if necessary. |
| crypt (1) | encrypt and decrypt files and streams. Note: the cipher used by Unix crypt has been broken and is not secure. |
| csv (3) | filter providing commands to process CSV data. |
| csvcat (3) | prepare and print a CSV file or stream for further processing. Comment lines that begin with a hash (#) and empty lines are eliminated. Lines that are distributed over multiple lines thru a backslash (\) at the end of the line are joined. White spaces before and after the field separator (;) are eliminated. |
| csvq (3) | csvq is a command line tool to operate CSV files. You can read, update, delete CSV records with SQL-like query. You can also execute multiple operations sequentially in managed transactions by passing a procedure or using the interactive shell. In the multiple operations, you can use variables, cursors, temporary tables, and other features. |
| csv2worksheet (3) | load a CSV file (comma separated file) into an existing Excel spreadsheet. It is possible to load the data into an existing worksheet (tab) or to create a new worksheet if the worksheet with the name specified does not exist. This is an easy method to create "nice looking" Excel reports based on plain ASCII data without the need to create Excel makros or to develop complicated programs that create the whole Excel sheet. The idea is to load the data into a separate "data" worksheet (tab) and to perform the calculations, graphics, statistics etc. in a worksheet (tab) that is not touched by the data load. With this method it is even possible to create Excel reports directly on Unix systems without manual intervention. |

D

| | |
|---------------------------|---|
| daemon_wrapper (1) | start commands that run as daemons out of the edrc/lib/daemon directory. |
| days (3) | evaluate days since January 1st of a given date. If no date is specified the number of days from January 1st until the current date is printed (this equals to date +%j on HP-UX systems). |
| dbrep (1) | execute reports against an Oracle database. |
| dig (1) | send domain name query packets to name servers. |

directories(3) expand a file list data stream with the overlying directories.

Example:

```
find /data/myApp -mtime 7 -print | directories |\
  cpio -pdvm /backup/myApp/20091212
```

dmidecode(3) **dmidecode** is a tool for dumping a computer's DMI (some say SMBIOS) table contents in a human-readable format. This table contains a description of the system's hardware components, as well as other useful pieces of information such as serial numbers and BIOS revision. Thanks to this table, you can retrieve this information without having to probe for the actual hardware.

dos2ux(3) convert ASCII file format between DOS and UNIX file format.

duvi(1) edit the **..du_index** file used by **du_report** to document your system.

du_report(1) create a system report (with filesystems, directories, sizes and directory descriptions).

E

ecronnext(1) show the commands that are scheduled next in the **crontab(4)** entries of the **crond(3)** bundled with WA2L/edrc.

ecrontab(1) add/edit **crontab(4)** entries of the **crond(3)** bundled with WA2L/edrc.

edrc(1m) this is the Enterprise Disaster Recovery Console. From here almost all recovery steps on UNIX can be performed.

edrcenv(1) print the official WA2L/edrc environment variables.

edrcinit(1m) handle services start/stop/restart, primarily for daemons included in WA2L/edrc, but other services can be handled as well. **edrcinit** supports to monitor and to restart a failed service based on the definitions made in the **edrcinit.cfg** config file.

edrclicense(4) GNU General Public License for the WA2L/edrc package.

| | |
|------------------------------|--|
| edrcman (1) | show manual pages of WA2L/edrc without the need to change the \$MANPATH by hand. |
| edrcpack (1m) | deprecated, use pack (1m) to create a software package of the application. |
| edrcperm.no_shell (3) | pseudo shell to prevent shell exits within edrc possible in vi (1), view (1) and more (1) if the invocation of the functionality <i>shell</i> is denied with the related setting in DENY_LIST in the edrc.cfg file. |
| edrcports (4) | directory of network ports and protocols used by default by the WA2L/edrc package. |
| edrcrevision (1) | This is not a command, but a documentation of the "revision history of WA2L/edrc" by Christian Walther. |
| edrcroot (3) | print installation root of WA2L/edrc. This command is identical with approot and will be removed as soon as all commands within EDRC use approot to resolve the installation root. |
| edrcsetup (1m) | This is not a command, but a documentation of "How to install WA2L/edrc (Quick Installation Guide)" by Christian Walther. |
| envpasswdstrip (1) | strip passwords from _env files. This might be useful if you like to share recovery scripts but don't like to hand out the passwords. |
| epub2pdf (1) | epub2pdf is a command-line tool that quickly generates PDF files from EPUB ebooks. It allows the user to specify page size, fonts, margins, and default paragraph alignment. See also http://epub2pdf.com/ |
| eshell (1) | start shell . This command is located in the pbin/ directory to enable the user to conveniently start the shell command in his own user context. |
| eterm (1) | <p>start edrc, asup, psup, shell etc. from pbin/ directory in a separate X-terminal window.</p> <p>The eterm command also supports to create Desktop icons in the Gnome desktop environment.</p> <p>This command is located in the pbin/ directory to enable the user to conveniently start it.</p> |

exiftool(1) Read and write meta information in files. **exiftool** has been written by Phil Harvey. See also: <http://www.sno.phy.queensu.ca/~phil/exiftool/>

expect(3) programmed dialog with interactive programs. **expect** is a program that "talks" to other interactive programs according to a script. Following the script, **expect** knows what can be expected from a program and what the correct response should be. An interpreted language provides branching and high-level control structures to direct the dialog. In addition, the user can take control and interact directly when desired, afterward returning control to the script.

F

fcreate(3) create a file if it does not exist. Optionally text can be written to the file created. The intention of this command is to use it to create lock files within shell scripts.

fields2swvi(3) convert a data stream to a correct swvi input sequence. **fields2swvi** is used by **apply2sw_inventory** internally. See also **swvi(1)** and **apply2sw_inventory(1)**.

filedist(1) distribute file(s) to a list of other systems.

filegrep(1) grep for a pattern in all underlying ASCII files. Files located in a **.sav** directory or files saved with the **sav(1)** command are excluded from the search. See also **scriptgrep(1)**.

filelink(1) create hardlinks between identical files in two directory trees.

filesize(3) print the size of a given file.

filewatch(3) watch the file size progress of a given running command.

fit(3) alias for **fit2width(3)**.

fit2width(3) fit a data stream to a certain width. This command can be used together with **tty_columns** to fit a stream to the current width of a terminal (tty).

fmatch(3) prints the values of the given key from the file csv database specified on command line or in the **\$FMATCH_DATA_FILE** environment variable.

| | |
|---------------------|---|
| fnmatch(3) | match a stream with a wildcard expression. |
| freespace(3) | print free space of the filesystem where the given directory is located on. |
| fssum(1) | summarize filesystems based on definitions in a configuration file. This command is used to create a filesystem summary grouped by product. |
| ftps(1) | secure FTP is an FTP program that allows for a secure connection to be made to an FTP server via a Secure Sockets Layer (SSL). This version of secure FTP supports both implicit and explicit SSL connections. See also http://www.glub.com/products/secureftp/ |

G

| | |
|-------------------------|---|
| gawk(1) | manual page of gawk , GNU Project's implementation of the AWK programming language. |
| gecos(3) | return the GECOS (=comment) field of a defined user account. |
| getfilesystem(3) | print the file system where the current working directory is related to. |
| gethostbyname(1) | host lookup using gethostbyname() system call. This tool is useful to check if your /etc/nsswitch.conf works properly. |
| getmountpoint(3) | print the mount point where the current working directory is related to. |
| gid(3) | return the group-id (GID) of an existing group. |
| glibc.version(3) | print version number of the glibc library. |
| group(3) | return the group name of an existing group for a given group-id. |
| gs(3) | <p>The gs command invokes Ghostscript, an interpreter of Adobe Systems' PostScript™ and Portable Document Format (PDF) languages.</p> <p>gs reads "<i>files</i>" in sequence and executes them as Ghostscript programs. After doing this, it reads further input from the standard input stream (normally the keyboard), interpreting each line separately and output to an output device (may be a file or an X11 window preview, see below).</p> |

The interpreter exits gracefully when it encounters the **quit** command (either in a file or from the keyboard), at end-of-file, or at an interrupt signal (such as **Control-C** at the keyboard).

gunzip(1) compress or expand files.

gzip(1) compress or expand files.

H

h2(1) H2 is a relational database management system written in Java. It can be embedded in Java applications or run in the client-server mode. The disk footprint (size of the jar file) is about 1.5 MB.

histlist(3) print a table of history files including the resolution of usage begin, usage end, history file size and history file name.
The begin can be resolved properly only if the history file names comply to the history file names as produced by the **shell**(1) command when started outside **edrc**.

homedir(3) return the homedir of a user.

hostaliases(3) return aliases for a given host.

hostlist(3) return a hostlist. This command is used to centralize the configuration of a list of hosts to a single configfile.

hostlistdat2cfg(3) convert a **hostlist.dat** file that has the structure

Example:

```
#
# Format:
#
# <CUSTOMER>; <ENVIRONMENT>; <GROUPS>; <OPTIONS>; <HOSTS>
#
ACME;DEVELOPMENT ;@APP ;;host-001 host-002;
ACME;DEVELOPMENT ;@DB ;;host-003;
ACME;TEST ;@APP ;;host-101 host-103;
ACME;PREPRODUCTION ;@APP ;;host-201 host-202 host-205;
ACME;PRODUCTION ;@APP ;;host-303 host-308 host-309;
```

into settings that can be used in the **hostlist.cfg** file.

This command will significantly simplify the definition of host lists of large environments and will make the more complicated constructs obsolete where using the **fmatch(3)** command in the **hostlist.cfg(4)** configuration file.

Despite the many options that the **hostlistdat2cfg** provides, normally only the **-m** and the **-o** options will be used in the **hostlist.cfg** config file. See also samples in the **edrc/var/samples/hostlist** directory.

| | |
|-------------------------|---|
| html2mht(3) | convert a HTML file to a single web archive file (MHT) that can be viewed using for instance the Microsoft Internet Explorer (TM) or an other MHT viewer. html2mht is currently able to convert local HTML files having the image files locally stored. CSS and JavaScripts have to be within the HTML file to get a good result when viewing the converted file. |
| hwinventory(1) | command to print a minimal set of inventory information needed for asset management. |
| I | |
| ident(3) | continuous output of characters received via stdin with a leading ident on each line. This command is depreciated, use indent(3) for new scripts. |
| indent(3) | continuous output of characters received via stdin with a leading ident on each line. |
| info(1) | read Info documents. |
| ini(1) | initialize the environment based on named definitions specified in etc/env . Each shell has its own ini.<shell> binary to initialize the environment based on the definitions. So it is not necessary to use different syntax for different shells, this is done by the ini.<shell> command. The interactive ini command is a shell alias pointing to the relating ini.<shell> command. The alias in the ksh shell therefore calls the ini.ksh command, a C-shell would call the ini.csh command. |
| input(3) | an interactive input command for use in own recovery scripts and contributed commands. |
| input_targets(3) | special purpose input to query lists of hosts. The prompt cannot be changed and is set to "Target hosts". The input and the optional command argument is a space separated list of hosts and hostgroups. If the input differs from the default the input prompt is repeated to allow verification and corrections. If no <i>target_list</i> is specified in input_targets <i>target_list</i> the default target list provided in the input prompt is as it is resolved by hostlist(3) . |

| | |
|-----------------------------|--|
| ipcalc (1) | ipcalc takes an IP address and netmask and calculates the resulting broadcast, network, Cisco wildcard mask, and host range. By giving a second netmask, you can design subnets and supernets. |
| ipsort (1) | with this command you can sort a data stream which has IP addresses in the first column. |
| is_config_byhand (3) | evaluate if a given file is listed in a config_byhand file. If so return 'True' else return 'False'. This command can be used in pre_exec and post_exec scripts within sysconfig . |
| is_existing (3) | check if a file/directory/link etc. is existing and accessible by the calling user. This command does not hang if for instance the NFS resource where the file/directory/link is located is not available. There is the possibility to specify a timeout value when the default of 0 second is not sufficient. |
| is_osid (3) | evaluate if the operating system id returned by osid (3) matches to one os-id specified by the -o option. This command is used to limit a script execution to supported operating systems only. Example: <pre>is_osid -s \$0 -o HP-11,Solaris exit \$?</pre> |
| is_permitted (3) | the main intention of this command is to check if a certain functionality is permitted for execution in contributed commands. In recovery scripts global functionality and script functionality can be checked for startup permission. This enables you to minimize possible by-passes of the denials defined in the DENY_LIST in the edrc config file edrc.cfg (4) when you use an edrc configuration for system management purposes. |
| is_running (3) | evaluate if a given cluster package is up. Return 'True' if it is up, otherwise return nothing, therefore this command can be used in crontab to run a certain line only if the specified cluster package is running on this host. Example: <pre>0 1 * * * ["`/opt/edrc/lib/is_running psoftprod`"] \ && /opt/psoft/bin/my_command</pre> |
| is_up (3) | evaluate if a system is up thru pinging it with a one second timeout. |
| is_user (3) | evaluate if the user that started the command matches to one username specified by the -u option. This command is used to limit a script execution to supported users only. |

is_weekend(3) return True if the current day is at the weekend, else return False.

is_writeable(3) check if write access to a file or a directory is possible. Return 'True'/'0' if write access is possible, otherwise return 'False'/'1'.

J

java(1) start **java** based on the setting in **java_wrapper.cfg**. The **java** command is started thru the **.java_wrapper**.

java_wrapper(1) wrap programs written in java that they can be started the same way at it would be native commands. The search order to the **java** command installed on the system or eventually distributed with WA2L/edrc has to be set in the configuration file **java_wrapper.cfg** if it is not found.

job(1) job sequencer. This command can be used to serialize jobs on a UNIX system. It has a similar user interface to the **at** command. The advantages of this command are multiple queue support, execution of jobs in the current user environment and a logging facility.

joblog(1) monitor the logfile of the **job** command.

jobstart(1) start jobs using a simple interface, where a job is identified by a **JOBNAME**.

jobwatch(1) monitor the state of jobs in a certain queue. The output is similar to the **top** command.

jq(3) **jq** is a lightweight and flexible command-line **JSON** processor.

See <https://stedolan.github.io/jq/manual/> and <https://stedolan.github.io/jq/tutorial/> for more information.

K

kalc(1) **kalc** is a programmable scientific calculator, using RPN (Reverse Polish Notation). It includes over 200 functions and a built-in help system. **kalc** works with real numbers, complex numbers and also integers in other numeric bases. Its has a complete programming language, with control-flow structures such as if and while. It has "unlimited" memory for you to store objects.

ksh(1) manual page of **ksh**, KornShell, a standard/restricted command and programming language.

| | |
|------------------------|---|
| kshell (1) | print path of the Korn- or Bourne Again shell available on the system. |
| ksh_wrapper (1) | wrap Korn- and Bourne Again shell scripts to avoid using <code>#!/bin/ksh</code> in the magic key. However, due to the fact that some scripts need Korn shell functionality either ksh or bash has to be installed in <code>/bin</code> , <code>/sbin</code> , <code>/usr/contrib/bin</code> or <code>/usr/local/bin</code> . |
| L | |
| lbanner (3) | print a ASCII banner as the banner (3) command does, but allow more options, as character size, fill character and banner orientation. |
| lcat (1) | alias for logcat (1). |
| leo (1) | leo is a command line interface to the german/english/french/spanish/... dictionary on http://dict.leo.org/ . It supports almost all features which the website supports, plus more. Results will be printed to the terminal. By default the searched key word will be highlighted. leo has been written by Thomas Linden <tom@daemon.de>. See also: http://search.cpan.org/~tlinden/ |
| lgcheckd (1m) | a daemon to check logfiles for patterns and to create a consolidated findings report. This is the more efficient and leaner variant of the logcheckd (1m) daemon command that operates on the same configuration file format, the same pattern files and interfaces as logcheckd (1m) for easy migration. |
| lgcpattern (3) | check/analyse/verify a logcheckd.pattern (4) file definition against a data stream. This utility helps to create the definitions in a pattern file. |
| lgrep (1) | alias for loggrep (1). |
| listtemp (3) | list temporary directories created with maketemp (3) and resolve if the related process that created the temp dir is still running. This command can be used for system housekeeping. |
| ll (1) | long file listing. This command calls internally ls -laF [options] and is to enable the user to use ll when calling rcmd even on operating systems where ll (1) is not provided. |

| | |
|-----------------------|---|
| llcomp (1) | detailed list of the contents of a compressed tar archive (.tar.gz , .tgz or .cpio.gz files) without decompressing it. |
| locate (1) | search for pattern in the locate database produced by updatedb (1m). |
| locations (1) | print a list of corporate locations with local time. |
| log (3) | write a log message to the logfile for use in contributed commands. |
| logcat (1) | cat selected time range (from/to) in log files having time stamps. To grep selected time range (from/to) in log files having time stamps use the loggrep (1) command. |
| logcheckd (1m) | a daemon to check logfiles for patterns and to create a consolidated findings report. The report can be mailed to a list of recipients. See also the new log checking daemon lgcheckd (1m). |
| logcut (1) | shorten logfiles to a number of rows. |
| loggrep (1) | grep selected time range (from/to) in log files having time stamps. |
| logrotate (1) | rotate logfiles and preserve a number of old logfiles. |
| logtail (1) | give a selection of common logfiles, which can be listed in \$HOME/.mylog_files , \$HOME/.log_files or in etc/log_files.cfg and start tail -f <selected_file> . |
| logview (1) | give a selection of common logfiles, which can be listed in \$HOME/.mylog_files , \$HOME/.log_files or in etc/log_files.cfg and start view <selected_file> . |
| lots (1m) | long term data save handling. |
| lotsctl (1) | lots control menu configuration short start for edrc . lotsctl provides an easy to use user interface to the lots (1m) command. In addition to the lots (1m) command, some reporting functionality is provided. See sat (1) for more information. |

| | |
|---------------------|--|
| lscomp (1) | list the contents of a compressed tar archive (.tar.gz , .tgz or .cpio.gz files) without decompressing it. |
| lscp (1) | <p>print a list of cp commands with files in the current working directory to the screen. This might be useful if you have to copy files and adjust the filenames of many recovery scripts.</p> <p>Example:</p> <pre>lscp > a; vi a; sh a; rm a</pre> |
| lsmv (1) | <p>print a list of mv commands with files in the current working directory to the screen. This might be useful if you have to adjust filenames of many recovery scripts.</p> <p>Example:</p> <pre>lsmv > a; vi a; sh a; rm a</pre> |
| lsof (1) | list open files. |
| lspm (1) | list installed perl modules including version and installation path. See also pmdesc (1). |
| lua (1) | lua is the standalone Lua interpreter. It loads and executes Lua programs, either in textual source form or in precompiled binary form. lua can be used as a batch interpreter and also interactively. |
| luac (1) | luac is the Lua compiler. It translates programs written in the Lua programming language into binary files containing precompiled chunks that can be later loaded and executed. |
| luaenv (3) | <p>print the environment used by lua to access the Lua modules bundled with WA2L/edrc. To set the environment prior to execution of lua, invoke:</p> <pre>eval `luaenv`</pre> |
| luarocks (1) | <p>start the luarocks command with settings tuned for the WA2L/edrc package and the Lua environment distributed with the package.</p> <p>See also https://luarocks.org/ and https://github.com/luarocks/luarocks/blob/main/docs/index.md for additional information.</p> |

| | |
|--------------------------------|--|
| lua version (3) | print the <i>major:minor</i> version of lua 3. |
| lua _wrapper (1) | wrap Lua scripts and programs to avoid using <code>#!/usr/bin/lua</code> or similar path names in the magic key. This enhances the portability of WA2L/edrc commands written in Lua to systems where Lua is not installed in the default install locations. lua has to be installed in edrc/bin/ , /bin/ , /sbin/ , /usr/bin/ , /usr/local/bin/ or /usr/contrib/bin/ . Further lua locations can be configured in the optional config file |
| lynx (1) | Lynx is a fully-featured World Wide Web (WWW) client for users running cursor-addressable, charactercell display devices (e.g., vt100 terminals, vt100 emulators running on Windows 95/NT or Macintoshes, or any other "curses-oriented" display). It will display hypertext markup language (HTML) documents containing links to files residing on the local system, as well as files residing on remote systems running Gopher, HTTP, FTP, WAIS, and NNTP servers. |
| M | |
| mail _file (1) | send a mail with a message content and file attachment(s). |
| maketemp (3) | create a unique non existing temporary directory with secure permissions. Use removetemp (3) to remove a temporary directory created with maketemp . |
| makeuser (1) | user friendly interface to mkuser . |
| mandir (4) | This is not a command, but a documentation of a "directory of manual page online resources" compiled by Christian Walther. |
| man2html (3) | converts a manual page as found in file (or stdin , in case no file argument, or the argument - , is given) from man-style nroff into HTML, and prints the result on stdout . |
| manvi (1) | edit manual pages in WA2L/edrc. It is possible to specify additional directories to be searched for manual pages in the configuration file manvi.cfg . |
| mc (1) | GNU Midnight Commander is a directory browser/file manager for Unix like operating systems. If the environment variable \$MCTERM is not set, the terminal emulation is set to <i>xterm</i> , what is fine for most cases. |
| mcedit (1) | Internal file editor of GNU Midnight Commander. If the environment variable \$MCTERM is not set, the terminal emulation is set to <i>xterm</i> , what is fine for most cases. |

| | |
|----------------------|---|
| mcview (1) | Internal file viewer of GNU Midnight Commander. If the environment variable \$MCTERM is not set, the terminal emulation is set to <i>xterm</i> , what is fine for most cases. |
| md2html (3) | convert markdown file to HTML. |
| md5string (3) | calculate MD5 checksums for strings line by line. |
| mediawiki (3) | get/put MediaWiki pages from/to a MediaWiki server. The communication does not need the MediaWiki API (api.php), therefore also MediaWiki servers running on older versions then 1.13 are supported. |
| mkuser (3) | create and initialize a new directory for a user. |
| msg (3) | write a message to the screen for use in own recovery scripts and contributed commands. |
| msmtp (3) | an SMTP client as an alternative to sendmail (8). |

N

| | |
|-----------------|---|
| name (1) | print description of a file or directory examined from the file header. |
| nano (1) | Nano's ANOther editor, an enhanced free Pico clone. nano is a small, free and friendly editor which aims to replace Pico, the default editor included in the non-free Pine package. Rather than just copying Pico's look and feel, nano also implements some missing (or disabled by default) features in Pico, such as "search and replace" and "go to line and column number". nano is included mainly into WA2L/edrc due to the possibility to run it in restricted mode (-R , or as rnano) and in a real viewer mode (-v) without the possibility of bypassing it, as view does allow with the :w! command. |

Example use of **nano** in recovery scripts:

as normal editor:

```
nano $file > `tty`
```

as editor with the restriction to edit only the given file:

```
rnano $file > `tty`
```

as a real read only viewer only allowing to view the given file:

```
rnano -v $file > `tty`
```

nc(3) **nc** is a simple Unix utility which reads and writes data across network connections, using TCP or UDP protocol.

nginx(3) a HTTP and reverse proxy server, mail proxy server. nginx (pronounced engine-x for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

nmap(1) **nmap** (Network Mapper) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. **nmap** uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

nologin(1) nologin shell to block users from login in to the system.

nping(1) **nping** is an open-source tool for network packet generation, response analysis and response time measurement. **nping** allows users to generate network packets of a wide range of protocols, letting them tune virtually any field of the protocol headers. While **nping** can be used as a simple ping utility to detect active hosts, it can also be used as a raw packet generator for network stack stress tests, ARP poisoning, Denial of Service attacks, route tracing, and other purposes.

O

omniutil(1m) a collection of OmniBack (OBII) enhancement utilities.

os_wrapper(1) wrap os dependent binaries to allow transparent shellscript programming.

osid(3) evaluate the systems operating system id.

osid.probe(3) library command to probe if the setting in **osid.dat** is compatible to the related compiled binaries in **bin/OSID/** and **lib/OSID/**.

osup(1) operation support configuration short start for **edrc**. See **sat**(1) for more information.

outex(1)

produce output of a selected logfile (without path) that is saved in **edrc/var/log/** that can be used in **contrib.doc(1m)** in the **OUTPUT-EXAMPLE** section.

The intension is to use the **outex logfile** command directly from within **vi** when editing a recovery script, as:

```
# D: Restore database.
#
~
~
~
: . !outex 2020-06-02_10.52.05__db_restore.log
```

This will include the contents of the logfile **edrc/var/log/2020-06-02_10.52.05__db_restore.log** (without logfile header) into the **vi** session. Each line will be prepended by the **# O:** documentation tag.

The specified *logfile* can also be a gzipped file, as: **2020-06-02_10.52.05__db_restore.log.gz**

Furthermore if the specified logfile **2020-06-02_10.52.05__db_restore.log** does not exist, but the **2020-06-02_10.52.05__db_restore.log.gz** does, the contents of the compressed file is included and vice versa.

P**pack(1m)**

create a shell archive (example: **edrc-1.4.08-200502150919.sh**) of WA2L/edrc that can be used to install WA2L/edrc on all supported operating systems. Log-files, backup files, lock files etc. are excluded from the package.

passwdcombine(1)

combine password files or password databases. This is useful if you have to apply a new set of users to a system and you like to prevent a change of certain users (as system users).

passwdsort(1)

sort a passwd file or data stream from stdin by UID.

passwdsyncd(1m)

a daemon to synchronize passwords over several systems. The daemon can be switched centrally to a maintenance mode. While in maintenance mode the synchronizations on all nodes taking part on a synchronization is on hold. The behavior of **passwdsyncd** can be configured in **passwdsyncd.cfg**.

passwdsyncd_apply(3)

apply passwords distributed with **passwdsyncd** to the target system.

patchinstall(1m)

patch an WA2L/edrc installation. **patchinstall** also cleans up corpses of older WA2L/edrc versions. All patched files and cleaned up corpses are backed up.

- pdfmetaedit(1)** GUI to edit meta data of a PDF file. This command was developed by "zarro". See also <http://zaro.github.io/pdf-metadata-editor/>.
- pdfscissors(1)** GUI to crop PDF file for eBook reader. This command was developed by Abdullah Al Mazed (Gagan). See also <http://pdfscissors.com/>.
- perlenv(3)** print the environment used by **perl** to access the Perl modules bundled with WA2L/edrc. To set the environment prior to execution of **perl**, invoke:
- ```
eval `perlenv`
```
- See also **perl\_modules(3)** for additional information.
- perlversion(3)** print perl version of **perl** interpreter found on the system. See also **perl\_wrapper(3)** for additional information.
- perl\_wrapper(1)** wrap Perl scripts to avoid using `#!/usr/bin/perl` or similar path names in the magic key. This enhances the portability of WA2L/edrc commands written in Perl to systems where perl is not installed in the default install locations. **perl** has to be installed in `/bin`, `/sbin`, `/usr/local/bin` or `/usr/contrib/bin`. Further **perl** locations can be configured in the config file **perl\_wrapper.cfg**.
- pf\_wrapper(1)** Wrap WA2L/edrc commands to be started thru **pfexec(1)** or **sudo(8)**.
- To use the commands in **pbin/** add the directory to the user's **\$PATH** variable:
- ```
[ /home/fred ]
[ fred@host-001 ][bash]: vi ~/.bashrc
:
27 # Add edrc/pbin to $PATH
28 PATH=~edrc/pbin:$PATH; export PATH
~
~
```
- pip(1)** start the **pip** command **pip3(1)** with settings tuned for the WA2L/edrc package and the Python3 virtual environment (**venv**) distributed with the package.
- pip3(1)** the **pip** command is the Python package manager. The **pip** command call within WA2L/edrc handles the virtual environment **venv** distributed with WA2L/edrc.
- pid(1)** select (grep) a given pattern out of the process list.

| | |
|-------------------------|--|
| pkgdir (1m) | This is not a command, but a documentation of a "directory of package/software handling commands" compiled by Christian Walther. |
| pkg_hostname (3) | return MC/ServiceGuard package name the current working directory is related to. If the directory is not related to a package the hostname is returned. This command can be used to set the user's prompt depending on the path the user is currently in. |
| pkzip (1) | compression utility. |
| pl (1) | pl is a program that produces plots and charts from data, and produces results that can be viewed on web pages, paper, slides, or interactively on the screen. A user developed script file may be supplied for greater flexibility and customization. pl may be executed from the command line or as a CGI program. |
| pmdesc (3) | List name, version, and description of all installed perl modules and PODs. See also lsbm (1). |
| pod2html (3) | Converts files from POD (Plain Old Documentation) format to HTML format. See also: https://perldoc.perl.org/perlpod and perlpod (4). |
| portscan (3) | scan a target system for active ports. It is possible to specify the minimum port where the scan should start and the maximum port number where the scan should stop. Ports that are active are returned to stdout . |
| print_header (3) | print a standard report header to the terminal. The output is limited to the current terminal width if the environment variable \$PRINT_FIT2WIDTH is not set to <i>False</i> . |
| print_index (3) | print a standard report column index to the terminal. The output is limited to the current terminal width if the environment variable \$PRINT_FIT2WIDTH is not set to <i>False</i> . |

Example:

```
cat<<EOM | print_index
ZIP;Postal Zip Code
CITY;Name of the City
STATE;State
CNT;Country Shortcut
COUNTRY; Country Full Name
EOM
```

COLUMN INDEX:

```
ZIP ..... Postal Zip Code      CNT ..... Country Shortcut
CITY ..... Name of the City     COUNTRY .. Country Full Name
STATE .... State
```

print_list(3)

the intention of this command is to print a list to a terminal in a good readable format. A csv file provided via pipe is printed to stdout. The first row is treated as header row. The width of all columns is dynamically adjusted to the row with containing the longest entry. Furthermore the output is limited to the current terminal width if the environment variable **\$PRINT_FIT2WIDTH** is not set to *False*. To select named columns prior to print the list with **print_list** use **select_columns(3)**.

Example 1:

```
cat<<EOM | print_list
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

| ZIP | CITY | STATE | CNT | COUNTRY |
|-------|-----------|-------|-----|--------------------------|
| 93117 | Goleta | CA | USA | United States of America |
| 8222 | Beringen | SH | CH | Switzerland |
| 8005 | Cape Town | WC | RSA | South Africa |

(3)

Example 2:

```
cat<<EOM | select_columns ";" "CNT;ZIP;CITY" | print_list
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

| CNT | ZIP | CITY |
|-----|-------|-----------|
| USA | 93117 | Goleta |
| CH | 8222 | Beringen |
| RSA | 8005 | Cape Town |

(3)

pscount(3)

count processes and evaluate if the situation correlates to a defined situation as received thru stdin.
This command is intended to be used to check if the correct number of needed processes for an application are running on the system.

Example:

```
cat << EOM | pscount
1; 1;Oracle Listener ;oracle;./tnslsnr LISTENER .+;
1;10;Oracle DB Writer;oracle;ora_dbw[0-9]_ACMEDB;
1; ;Oracle Connects ;oracle;oracleACMEDB .+;
EOM
```

| DESCRIPTION | MIN | MAX | CURRENT | STATE |
|------------------|-----|-----|---------|-------|
| ----- | --- | --- | ----- | ----- |
| Oracle Listener | 1 | 1 | 1 | OK |
| Oracle DB Writer | 1 | 10 | 2 | OK |
| Oracle Connects | 1 | | 0 | FAIL |
| *TOTAL* | 3 | 11 | 3 | FAIL |
| (4) | | | | |

- psjoin(3)

join concatenates several PostScript files and generate a single PostScript document. The output, concatenated PostScript document, will be written to the standard output.

The input PostScript files must comply with the DSC (Document Structuring Convention). **psjoin** can fail to work depends to the input PostScript file or combination of the input PostScript files.
- pslist(3)

list and select processes in a cross operating system compatible format. When not invoked with the **-l** (= long output) option, only a list of the process-ids of the selected processes is printed.
- ps2pdf(3)

convert PostScript files to PDF.
- pstree(1)

print a process tree.
- psup(1)

production support configuration short start for **edrc** . See **sat(1)** for more information.
- purgetemp(3)

remove all temporary directories in a base dir that where created by **maketemp(3)** and have the state **IS_ACTIVE=False**.
- pwcrypt(1)

encrypt a plain password.
- pwsafe(3)

this command that can be called from outside of **edrc(1m)** allows also to access the password safe of a recovery script tree.

In this case the master password has to be provided thru the environment variable **\$PWSAFE_MASTERPASSWORD**.

Of course one using this command should be careful about how to handle the master password.

See **contrib.edrc(1m)**, **contrib.pwsafe(1m)** and **pwsafe(3)** for more information.

pythonenv(3) print the environment used by **python** to access the Python modules bundled with WA2L/edrc. To set the environment prior to execution of **python3**, invoke:

```
eval `pythonenv`
```

pythonversion(3) print the *major.minor* version of **python3**.

python_wrapper(1) wrap Python scripts to avoid using `#!/usr/bin/python3` or similar path names in the magic key. This enhances the portability of WA2L/edrc commands written in Python to systems where Python is not installed in the default install locations. **python3** has to be installed in `/bin`, `/sbin`, `/usr/bin`, `/usr/local/bin` or `/usr/contrib/bin`. Further **python** locations can be configured in the optional config file **python_wrapper.cfg**.

R

random(3) return a random number or a random item of a specified item list.

rcat(1) display (**cat**) remote- or local file(s).

rcomm(1) compare (**comm**) remote- or local file(s).

rcmd(1) run a command or a set of commands on a set of hosts. You can specify the command(s) in the command line, if you don't you are prompted for it.

rdiff(1) show differences (**diff**) of remote- or local file(s).

readline(3) read long input lines from **stdin** and return the input to **stdout**. The maximum length of the input is 10240 characters (10 kBytes), input longer then this maximum is ignored and truncated.

regexintro(4) This is not a command, but a documentation that gives an "Introduction to Regular Expression Usage" compiled by Christian Walther.

resolve_targetlist(3) resolve a targetlist which may consist of a list of hosts and hostgroups.

rel2abs(3) convert a relative filename to an absolute.

| | |
|-------------------------|---|
| remote_copy (3) | wrapper for remote copies. This command enables you to write scripts independent of the current security policy of your systems. Internally remote_copy uses r cp or s cp dependent of the configuration settings. Furthermore this command provides a caching mechanism which remembers successful connection modes to speed up future connections. |
| remote_shell (3) | wrapper for remote shells. This command enables you to write scripts independent of the current security policy of your systems. Internally remote_shell uses remsh , rsh or ssh dependent of the configuration settings. Furthermore this command provides a caching mechanism which remembers successful connection modes to speed up future connections. |
| removetemp (3) | remove a temporary directory created with maketemp (3). |
| repeat (3) | the repeat command re-executes the single subsequent command for count number of times. |
| revision (1) | print the most recent revision entry of a file examined from the file header. |
| rl (1) | shortcut for the rlogin command. See rlogin (1) manpage for additional information. |
| rnano (1) | Restricted mode for Nano's ANOther editor (nano), an enhanced free Pico clone. rnano is a restricted version of nano, which only edits specific files and doesn't allow the user access to the filesystem or a command shell. See example usage in intro of nano above. |
| role_option (3) | return the value of a certain option related to a role in an environment. This function is used to support a role based user creation model based on role templates which can be defined in the relating configuration file role_option.cfg . |
| rosid (3) | evaluate the systems operating system id of a remote system. |
| rsat (1) | start sat on a remote system. The list of hosts provided can be specified in the configuration file rsat.cfg . |
| rssh (1) | connect to a remote system using ssh (1). When invoking the rssh command the remote user and the remote host is queried interactively, where the last chosen user- and host-name is provided as default |

input.

When a symlink to the **rssh** command in the format **rssh-user@host** is created, the newly created command variant will connect directly to the *host* with the *user* and the remote system will query for the password.

The main purpose for this is to create a pseudo user (without password) and add the command variant **rssh-user@host** to the shell field of the **/etc/passwd** field to allow to use a local system to be used as a direct terminal to a remote system, as for example:

```
acme:x:1291:1291:acme login terminal:/tmp:/home/fred/bin/rssh-f
```

This would allow you to enter *acme* as login on the local system that will connect to the remote *host* where the login password of the *user* will be queried.

rsync(1)

rsync is a program that behaves in much the same way that **rcp** does, but has many more options and uses the **rsync** remote-update protocol to greatly speed up file transfers when the destination file is being updated.

S

sat(1)

short start of **edrc** with an other configuration which points to an own script tree. Internally **sat** calls **edrc -c edrc.sat.cfg**. **sat** stands for "system administration tool". If additional short starts are needed, create a symlink from the new short start command to **sat**.

Example:

```
cd ~edrc/bin
ln -s sat new_shortstart
```

When starting **new_shortstart** internally **edrc -c edrc.new_shortstart.cfg** will be called.

Currently the following short start commands are distributed with WA2L/edrc: **asup** (Application **S**upport Configuration), **psup** (Production **S**upport Configuration), **osup** (Operation **S**upport Configuration), **lotsctl** (Long Term data Save Control Configuration).

sav(1)

save a file to a backup with preserving the filepermissions and file dates.

screen(1)

screen is a full-screen window manager that multiplexes a physical terminal between several processes (typically interactive shells). Each virtual terminal provides the functions of a DEC VT100 terminal and, in addition, several control functions from the ISO 6429 (ECMA 48, ANSI X3.64) and ISO 2022 standards (e.g. insert/delete line and support for multiple character sets). There is a scroll-back history buffer for each virtual terminal and a copy-and-paste mechanism that allows moving text regions between windows.

| | |
|-----------------------------|---|
| scriptextract (3) | extract scripts distributed with the distribute EDRC command. |
| scriptgrep (1) | grep all recovery scripts for a pattern. Files in .sav directories and files not following the name convention *:* are excluded from the search. See also filegrep (1). |
| scriptheadersync (1) | synchronize the header entries in recovery scripts with the actual filename of the script. The script starts in the current working directory and searches all underlying directories for recovery scripts matching the filename convention: <i><menu-point>:<scriptname></i> . |
| scriptmenupath (3) | return recovery script menu path when called from a recovery script or _env file in the form 'menu -> submenu -> subsubmenu -> menupoint'. |
| scriptrevision (3) | print the revision number of a (script)file following the [##] revision notation as also resolved by revision (1). |
| scriptsequence (3) | <p>check if recovery scripts are called in ascending sequence. If not a message is printed or the script is aborted. scriptsequence must be called from a recovery script or _env file to give the user more awareness if the sequence is broken.</p> <p>Usage example (query to abort or not):</p> <pre>scriptsequence exit</pre> <p>Usage example (inform only):</p> <pre>scriptsequence -a inform exit</pre> |
| scripttitle (3) | print recovery script title (menupoint, description and duration as defined in doc tag) when called from a recovery script or _env file to give the user more awareness which menu point is started. |
| seconds (3) | calculate seconds since the Epoch for a given date and time. Use timer to evaluate the seconds since the Epoch for the current moment. |
| sectioncat (3) | cat section (<i>[SECTION]</i>) of a text file. |
| sed1line (1) | This is not a command, but a documentation of "USEFUL ONE-LINE SCRIPTS FOR SED (Unix stream editor)" compiled by Eric Pemet. |

select_columns(3) filter to select named columns that are separated by a field separator from **stdin** and print the selected columns to **stdout**. The first row of the data stream must contain the row header. Columns that do not exist, are printed as empty columns.

Example:

```
cat<<EOM | select_columns ";" "CNT;ZIP;CITY"
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

```
CNT;ZIP;CITY;
USA;93117;Goleta;
CH;8222;Beringen;
RSA;8222;Cape Town;
```

server_environment(3)

return a name for the environment of the server where running the command. In the configuration file **server_environment.cfg** it can be defined which server relates to what environment. The **server_environment** command can be used to develop scripts which act specific on different environments without hardcoding hostnames into the scripts.

Example:

```
case `server_environment` in
    TEST)
        Max_load=100
        Mail_to=fred.developer@acme.com
        ;;
    PREPRODUCTION)
        Max_load=200
        Mail_to=barney.verifier@acme.com
        ;;
    PRODUCTION)
        Max_load=250
        Mail_to=wilma.production@acme.com
        ;;
    unknown)
        echo "server environment unknown, aborting."
        exit 1
        ;;
esac
```

sh(1)

manual page of **sh**, command interpreter (shell).

shell(1)

the **shell** command provides a slightly reduced environment as the **shell** command started within **edrc**. The differences to the **shell** start in **edrc** is, that the environment variables **\$EDRC_SESSION**, **\$EDRC_SCRIPTS_BASEDIR**, **\$EDRC_ENV**, **\$EDRC_CONFIGFILE**, **\$EDRC_RECOVERYTIME**,

\$EDRC_DIST_USER, **\$EDRC_NLS_DATE_FORMAT**, **\$EDRC_NLS_LANG**, **\$NLS_DATE_FORMAT** and **\$NLS_LANG** are not set. Call the **edrcenv**(1) command to display the official WA2L/edrc environment variables.

shellinaboxd(3)

publish command line shell through AJAX web interface.

In WA2L/edrc the **shellinaboxd** server process is configured in the **edrc/etc/shellinaboxd.cfg** configuration file and started thru **edrcinit**(1m). Use the command:

```
edrcinit start shellinaboxd
```

to start the shell web server process.

In the default configuration, the user then can connect to the address **https://servername:8806** and work with the command line shell thru the web browser without the need of installing a terminal emulation program locally. A connect to **http://servername:8806** is redirected to the **https** address.

See also: <http://code.google.com/p/shellinabox/> and <http://www.tecmint.com/shell-in-a-box-a-web-based-ssh-terminal-to-access-remote-linux-servers/>.

shlib(3)

print the environment variables needed by the dynamic linker **ld**(1) to locate the shared libraries. If a script need to set this environment variables (**\$LD_LIBRARY_PATH** and **\$SHLIB_PATH**), use

```
eval `shlib`
```

instead of setting them by your own. This ensures, that all scripts will continue to run, even when the directory structure in WA2L/edrc changes.

sortc(3)

sorting data in the "C" locale. Internally the **sort**(1) command is called.

sortv(3)

Provide a version sort functionality as provided by the **sort -V** command to operating systems, where **sort**(1) lacks this functionality.

sparse(3)

filter to identify sparse files. The filelist received thru **stdin** is searched for sparse files and the file names of all identified sparse files, or all non-sparse file names (**-v**), are printed to **stdout**. This enables to use the output directly for further processing thru a pipe for instance.

Examples:

Find sparse files in the current directory:

```
find . -print | sparse
```

Copy all sparse files:

```
find . -print | sparse | cpio -pdvm --sparse /destination
```

Copy all non-sparse files:

```
find . -print | sparse -v | cpio -pdvm /destination
```

- splitvt(1)** This program splits the screen into two windows, one above the other, and runs a shell in each one.
- ssh-exec(1)** execute commands provided thru **stdin** on a remote system using **ssh**.
- Example:
- ```
ssh-exec -u fred -t acme-007 <<EOC
date
uptime
uname -a
EOC
```
- ssh-exec** also supports to remote control commands using **expect(3)** without the need to write whole **expect** "wrapper" scripts or without even writing **expect** code.
- ssh-keyadd(1m)** Add SSH public keys to a users (normally root's) **authorized\_key** file(s). Furthermore the SSH daemon configuration file is modified that way that it is allowed to connect as root. To make the configuration change active, the HUP signal is sent to the SSH daemon.
- sshpass(1)** **sshpass** is a utility designed for running **ssh** using the mode referred to as "keyboard-interactive" password authentication, but in non-interactive mode.
- sqlite(1)** SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. It is used in countless desktop computer applications as well as consumer electronic devices including cellphones, PDAs, and MP3 players. The source code for SQLite is in the public domain. See <http://www.sqlite.org/docsrc> for more information.
- stat(1)** This command returns the file status information returned by the **stat(2)** system call.
- To print the modification time in a consistent format use
- ```
timer `stat -s mtime -f my_file`
```
- The advantage of using this command instead of
- ```
ls -al my_file
```
- is, that the output format does not change over time, as the **ls** does.
- strace(1)** trace system calls on Linux. This command is called internally, if the **truss** command is invoked on Linux.

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>streamcat(3)</b> | continuous unbuffered output of an input stream received thru <b>stdin</b> to <b>stdout</b> .                                                                                                                                                                                                                                                                                                                               |
| <b>svi(1)</b>       | edit the file in a <b>sysconfig</b> repository which is related to the local (active) file on the system. This command is equal to the <b>sysvi</b> command.                                                                                                                                                                                                                                                                |
| <b>sw_report(1)</b> | create a software inventory report of a system. This command takes the <b>..sw_inventory</b> files as a base for information.                                                                                                                                                                                                                                                                                               |
| <b>swvi(1)</b>      | edit the <b>..sw_inventory</b> file used by <b>sw_report</b> to create a software inventory of your system.                                                                                                                                                                                                                                                                                                                 |
| <b>symlink(1)</b>   | create a symbolic link. In addition to the <b>ln -s</b> command <b>symlink</b> removes an existing symbolic link or file with the same name prior to the symbolic link creation. If the <b>-l</b> option is specified <b>symlink</b> only proceeds if the file is an existing symbolic link; the intention of this option is to use <b>symlink</b> to switch an existing symbolic link from one to another target.          |
| <b>sys(1)</b>       | System configuration handling and some automated EDRC tasks, as WA2L/edrc package distribution and patch installation. See <b>sys(1)</b> for more information.                                                                                                                                                                                                                                                              |
| <b>sysconfig(1)</b> | configure a system with a prepared set of files with a single command.                                                                                                                                                                                                                                                                                                                                                      |
| <b>sysinfo(1)</b>   | prints information on overall system statistics. If called with an option to specify a dedicated information, the current time, the number of seconds since the Epoch and the relating information is printed separated by a deliminators on a single line. The main intention of the command was to print the system uptime in a more computable fashion ( <b>sysinfo -u</b> ) then returned by the <b>uptime</b> command. |
| <b>syscp(1)</b>     | copy an (active) file on the system to the related <b>sysconfig</b> repository.                                                                                                                                                                                                                                                                                                                                             |
| <b>sysdiff(1)</b>   | list if an (active) file on the system applied with <b>sysconfig</b> differs to the file(s) in the related repository. The files are considered as different if the content or the filepermissions differ. To obtain a plain list of files that differ from the applied file, redirect the output to <b>stderr</b> .                                                                                                        |
| <b>syspoll(1)</b>   | poll systems and execute a query. The main purpose is to centrally control report data collection.                                                                                                                                                                                                                                                                                                                          |

**sysvi**(1) edit the file in a **sysconfig** repository which is related to the local (active) file on the system. This command is equal to the **svi** command.

**sys2html**(1m) create a 'System Configuration' HTML report similar to **cfg2html**(1m) from remote systems without the need to install it on the systems where the configuration shall be collected.

## T

**tcpdump**(1) dump traffic on a network.

**textblock**(3) format a stream into a justified text block with an optional hanging first line.

**textcolor**(3) set the terminal text color. The *ATTRIBUTE* option can be set to **RESET**, **BRIGHT**, **DIM**, **UNDERLINE**, **BLINK**, **REVERSE** or **HIDDEN**. The *FOREGROUND* and *BACKGROUND* options can be set to **BLACK**, **RED**, **GREEN**, **YELLOW**, **BLUE**, **MAGENTA**, **CYAN** or **WHITE**.

**thttpd**(3) **thttpd** is a simple, small, fast, and secure HTTP server.

**timer**(1) return the number of seconds since the Epoch or return the date based on the given number of seconds since the Epoch. Use **seconds** to evaluate the number of seconds for a specific date.

**timezone**(3) return the timezone as set in the **timezone.cfg** file or as defined system wide.

**title**(1) set the title of the terminal window. There are two predefined titles that can be chosen using the **short** or **long** option. It is also possible to define an own title text.

**today**(3) return the date of today. It is also possible to specify the format of the return value as known from the **date**(1) command. This command can be substituted directly by **date** and does exist, to have an equivalent command to **tomorrow**(3) and **yesterday**(3) returning the value in the identical format.

**tolower**(3) return a given string in lowercase.

**tomorrow**(3) return the date of tomorrow. It is also possible to specify the format of the return value as known from the **date**(1) command.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>top</b> (1)           | displays the top processes on the system and periodically updates the information. Raw CPU percentage is used to rank the processes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>toupper</b> (3)       | return a given string in uppercase.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>tpl</b> (1)           | select and print a template file to <b>stdout</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>tput_examples</b> (3) | examples on the use of the <b>tput</b> (1) command to control terminal outputs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>traceroute</b> (1m)   | print the route packets take to network host.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>trash</b> (1)         | remove file(s) and create a <b>&lt;filename&gt;.TRASHED</b> file that contains information about the removal and the possibility of a restore of the file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>truss</b> (1)         | <b>truss</b> traces the system calls a process invokes and the signals it receives. It displays arguments in a symbolic way, shows the first bytes of read and write buffers as well as signal information when available. <b>tusc</b> can attach to live processes by providing PIDs (process IDs) as argument(s).                                                                                                                                                                                                                                                                                                                                                                                |
| <b>tscat</b> (3)         | filter to add a timestamp and a filename to a stream.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>tsize</b> (3)         | <p>this command returns the current width and height of the terminal in the <b>COLUMNS</b> and <b>LINES</b> variables. This output can be used directly to initialize the environment variables using the <b>eval</b> command:</p> <pre>eval `tsize`</pre> <p>The output of <b>tsize</b> is identical to the <b>resize</b>(1) command as provided when X11 is installed. As long as <b>resize</b> is available on the system it is used internally to resolve the terminal size, if not, a fallback exists to evaluate the terminal without the use of <b>resize</b>. Therefore it is not imperative to have X11 installed on the system when using <b>tsize</b> to resolve the terminal size.</p> |
| <b>ttyplay</b> (1)       | player of the tty session recorded by <b>ttyrec</b> (1).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>ttyrec</b> (1)        | a tty recorder.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>ttytime</b> (1)       | print the time of the recorded session data by <b>ttyrec</b> (1).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**tty\_columns(3)** return the current width of the terminal (tty).

**tty\_variable(3)** return the current value of a terminal (tty) setting.

**tzdump(3)** dump the contents of a zoneinfo file.

## U

**uid(3)** return the user-id (UID) of an existing user.

**uncbz(1)** unpack b2zipped cpio file(s) to the current directory. The source file remains packed.

**uncbz2(1)** unpack b2zipped cpio file(s) to the current directory. The source file remains packed.

**uncgz(1)** unpack gzipped cpio file(s) to the current directory. The source file remains packed.

**unzip(1)** unpack zipped cpio file(s) to the current directory. The source file remains packed.

**unczst(1)** unpack zstd-compressed cpio file(s) to the current directory. The source file remains packed.

**unczstd(1)** unpack zstd-compressed cpio file(s) to the current directory. The source file remains packed.

**undeb(1)** unpack file(s) of a rpm file(s) to the current directory. The source file remains packed.

**uniqpath(3)** removes duplicate entries in a colon separated string without changing the field order. This command can be used to tidy up **\$PATH**, **\$MANPATH** or **\$USAGE\_PATH** settings.

**unrpm(1)** unpack file(s) of a rpm file(s) to the current directory. The source file remains packed.

|                     |                                                                                                                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>untbz(1)</b>     | unpack b2zipped tar file(s) to the current directory. The source file remains packed.                                                                                                                                                                                                                 |
| <b>untbz2(1)</b>    | unpack b2zipped tar file(s) to the current directory. The source file remains packed.                                                                                                                                                                                                                 |
| <b>untgz(1)</b>     | unpack gzipped tar file(s) to the current directory. The source file remains packed.                                                                                                                                                                                                                  |
| <b>untrash(1)</b>   | restore file(s) removed by <b>trash</b> .                                                                                                                                                                                                                                                             |
| <b>untxz(1)</b>     | unpack xz-ed tar file(s) to the current directory. The source file remains packed.                                                                                                                                                                                                                    |
| <b>untzst(1)</b>    | unpack zstd-compressed tar file(s) to the current directory. The source file remains packed.                                                                                                                                                                                                          |
| <b>untzstd(1)</b>   | unpack zstd-compressed tar file(s) to the current directory. The source file remains packed.                                                                                                                                                                                                          |
| <b>unzip(1)</b>     | <b>unzip</b> will list, test, or extract files from a ZIP archive, commonly found on MS-DOS systems.                                                                                                                                                                                                  |
| <b>unzstd(1)</b>    | expand files.                                                                                                                                                                                                                                                                                         |
| <b>updatedb(1m)</b> | create the locate database to be queried by <b>locate(1)</b> .                                                                                                                                                                                                                                        |
| <b>usage(1)</b>     | print a short usage of non standard commands available in WA2L/edrc.                                                                                                                                                                                                                                  |
| <b>user(3)</b>      | return the user name of an existing user for a given user-id.                                                                                                                                                                                                                                         |
| <b>user_info(1)</b> | print information of a user defined on the system. This is an interface to read with one command in a structured matter from several files which are needed to finally define a user on a Unix/Linux system (passwd, group, samba user map, ftpusers, trusted system settings where applicable, ...). |

**uudecode(1)** **uudecode** transforms uuencoded files into their original form.

**uuencode(1)** **uuencode** is used to create an ASCII representation of a file that can be sent over channels that may otherwise corrupt the data. Specifically, email cannot handle binary data and will often even insert a character when the six character sequence "Orom " is seen.

**ux2dos(3)** convert ASCII file format between UNIX and DOS file format.

## V

**vadsp(1)** display additional information of a VA-7410 of HP on HP Unix systems attached to a VirtualArray.

**vcap(1), vmore(1), vdiff(1), vgrep(1), vlist(1), vpurge(1), vrestore(1), vsav(1)**  
the **v(sav|cat|more|diff|grep|list|purge|restore)** commands enable to save, print, compare, grep, list, purge and restore *versions* of a given *file*.

**vsdfml(3)** very simple document formatting language. The main purpose is to highlight documentation files on terminal output when writing recovery scripts whose purpose is to display only information and not to execute commands. To benefit from **vsdfml** you have to add it to the magic key of your information "recovery script": **#!/bin/sh vsdfml**. Lines starting with a hash ( # ) are considered as comments and are not printed. Repetitive empty lines are reduced to one line.

**vvi(1)** versioned file editor using **vi(1)** respectively the editor set in **\$EDITOR**.  
Prior to editing the given files each file is saved using **vsav(1)**.

## W

**watchdog(1)** a watchdog to check a certain condition. If the checked condition matches, bite.

**wa2lc(3)** directory of additional C include files provided in the WA2L/edrc package.

**wa2ledrc\_edrcapi(1)** web application **WA2L/edrc:edrcapi** (EDRC REST API) to serve WA2L/edrc information in API fashion.

**wa2ledrc\_report(1)** web application **WA2L/edrc:report** (OPERATING SYSTEM REPORT PORTAL) to serve mainly static operating system reports that are saved in a simple directory structure ( **<Customer>/<Report\_Name>/<Report\_file>** ). A main goal is to provide a web interface without the need of extensive configuration. However, a user administration web interface is included.

The portal can be provided for a single customer e.g. when running directly in the



customer's environment, or for a consolidated multiple customer view for system administration personnel.

#### **wa2ledrc\_shellinaboxd(1)**

web application **WA2L/edrc:shellinaboxd** to serve WA2L/edrc command line over Web browser.

#### **wget(1)**

GNU Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies.

#### **whatis(1)**

search the whatis database for complete words.

#### **whereami(1)**

print the server environment where logged on. In addition the structured user definition information of the current user is printed.

#### **whoisin(1)**

print a list of users who are logged in on certain (remote) hosts.

#### **wmic(1)**

client to use the Windows Management Instrumentation from Linux.

#### **wine(1)**

client to execute commands remotely on Windows from Linux.

#### **woist(1)**

search a path for a file.

### **X**

#### **xbdf(1m)**

a platform independent version of the **bdf** respectively **df -k** command. Line breaks are removed, as experienced on HP-UX in certain circumstances.

#### **xlog(1)**

write date from stdin to a logfile. Each line is preceded with a date, time, sequence and tag entry. A common use might be in crontab instead of piping outputs to **/dev/null**.

Example:

```
a_command 2>&1 | xlog -t a_cmd -f /var/adm/log/a_command.log
```

#### **xml2csv(3)**

convert XML schema to CSV.

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>xml2json</b> (3)  | convert XML schema to JSON.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>xmore</b> (1)     | display file(s) page by page in WA2L/edrc recovery scripts. The <b>more</b> (1) or <b>less</b> (1) command do not stop for user input after displaying a page if used in a recovery scripts.                                                                                                                                                                                                                                                                                              |
| <b>xpid</b> (1)      | <p>select processes from the process command part of the process list. This is a convenience command to the <b>pslist</b>(1) command.</p> <p>Internally the <b>xpid</b> command calls currently <b>pslist -l -c '.*selection.*'</b> .</p> <p>The <b>xpid</b> command is intended for manual command line use only, for script use the <b>pslist</b> command should be used because the output of the <b>xpid</b> command is not intended for further processing and might be changed.</p> |
| <b>xtee</b> (1)      | an enhanced version of the standard <b>tee</b> command. This one appends the standard output with extra data and session columns to the logfile. It is planned to use this command by <b>edrc</b> internally to save the standard output of the recovery scripts to the logfiles.                                                                                                                                                                                                         |
| <b>Y</b>             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>yesterday</b> (3) | return the date of yesterday. It is also possible to specify the format of the return value as known from the <b>date</b> (1) command.                                                                                                                                                                                                                                                                                                                                                    |
| <b>ypxfr_all</b> (1) | transfer NIS maps from a NIS master server.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Z</b>             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>zcat</b> (1)      | <b>zcat</b> is identical to <b>gzip -c</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>zdiff</b> (1)     | compare compressed files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>zgrep</b> (1)     | search possibly compressed files for a regular expression.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>zip</b> (1)       | <b>zip</b> is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows NT, Minix, Atari and Macintosh, Amiga and Acorn RISC OS.                                                                                                                                                                                                                                                                                                                                       |
| <b>zless</b> (1)     | file perusal filter for crt viewing of compressed text.                                                                                                                                                                                                                                                                                                                                                                                                                                   |

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| <b>zmore(1)</b> | file perusal filter for crt viewing of compressed text. |
| <b>zstd(1)</b>  | compress or expand files.                               |

## FILES

The **name(1)** command can be used to list a short description of files and directories.

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>.ssh</b>              | DO NOT SAVE *ANY* DATA INTO THIS DIRECTORY.<br><br>This directory contains *no* <b>ssh</b> keys, <b>ssh</b> authorized key files or <b>ssh</b> configuration files. The <b>ssh</b> files are saved in the <b>var/connection/</b> structure. See below for more information.<br><br>The reason why this directory exists is, that on certain operating systems under special conditions (as HP-UX 11.11 having the random number generator patch not installed), the connection in OpenSSH mode fails when using the commands <b>rcmd</b> , <b>filedist</b> , <b>remote_copy</b> , <b>remote_shell</b> or when establishing trunks in <b>edrc</b> . |
| <b>pbins/</b>            | binary directory for commands that could be started with elevated permissions. This directory only contains symlinks to the <b>.pf_wrapper</b> . See <b>pf_wrapper(1)</b> for additional information.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>sbin/</b>             | this is the location of the <b>edrc</b> command. This directory should not be included in \$PATH.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>bin/</b>              | some general use commands. This directory is in \$PATH.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>bin/&lt;OSID&gt;/</b> | operating system dependent commands. These commands are mostly compiled for the designated operating system. The command call occurs via the <b>os_wrapper</b> command and never via a \$PATH enhancement or a direct call.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>contrib/edrc/</b>     | contributed commands to <b>edrc</b> . Do not change the <b>edrc</b> program if you miss a general command. Create the command in this directory and you will be able to invoke it as it is an <b>edrc</b> internal command. This will protect you from losing your work after an upgrade of <b>edrc</b> and will keep the <b>edrc</b> command itself stable.                                                                                                                                                                                                                                                                                       |
| <b>doc/</b>              | some printable documentation, including all man pages, in PDF, PostScript and HTML format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>etc/</b>              | configuration files                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>lib/</b>              | commands which are mostly used within scripts. Some of them require some environment settings. This directory is in \$PATH.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>lib/&lt;OSID&gt;/</b> | operating system dependent library commands. These commands are mostly compiled for the designated operating system. The command call occurs via the <b>os_wrapper</b> command and never via a \$PATH enhancement or a direct call.                                                                                                                                                                                                                                                                                                                                                                                                                |

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>lib/&lt;OSID&gt;/libs/</b> | shared libraries needed by the compiled operating system dependent commands located in <b>bin/&lt;OSID&gt;/</b> and <b>lib/&lt;OSID&gt;/</b> started thru the <b>os_wrapper</b> . The <b>os_wrapper(3)</b> ensures that this directory is added to the <b>\$LD_LIBRARY_PATH</b> and the <b>\$SHLIB_PATH</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>lib/daemon/</b>            | Korn shell scripts called via <b>daemon_wrapper(1)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>lib/dbrep/</b>             | SQL reports used by <b>dbrep(1)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>lib/edrc/</b>              | library functions called by the <b>edrc(1m)</b> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>lib/icons/</b>             | icon images.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>lib/java/</b>              | location of Java <b>.class</b> , <b>.jar</b> , <b>.class.javaopt</b> and <b>.jar.javaopt</b> files. The Java programs are called via the <b>java_wrapper(1)</b> . Therefore when calling java commands in WA2L/edrc this is transparent to the user and the command can be called by simply entering <b>my_command my_options</b> , there is no need to call <b>java java_options my_command my_options</b> as it is normally the case. Using the <b>java_wrapper(1)</b> there is no need to write a start script for each Java program. If a certain Java program needs special options passed to java, those options can be specified in the config file <b>lib/java/filename.class.javaopt</b> or <b>lib/java/filename.jar.javaopt</b> with the <b>JAVA_OPTIONS</b> setting. |
| <b>lib/ksh/</b>               | Korn shell scripts called via <b>ksh_wrapper(1)</b> . In WA2L/edrc Korn shell scripts are not executed via the magic key <b>#!/bin/ksh</b> due to the fact that this is less portable than Bourne shell scripts. Shell locations can be configured in the configuration file <b>ksh_wrapper.cfg</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>lib/logcheckd/</b>         | files used by the <b>logcheckd(1m)</b> when generating LogCheck reports.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>lib/lua/</b>               | Lua scripts called via <b>lua_wrapper(1)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>lib/lua/lum/</b>           | additional Lua modules used by the Lua scripts/programs started thru the <b>lua_wrapper(1)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>lib/pack/</b>              | library functions used by <b>pack(1m)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>lib/perl/</b>              | Perl scripts called via <b>perl_wrapper(1)</b> . In WA2L/edrc Perl scripts are not executed via the magic key <b>#!/usr/bin/perl</b> due to the fact that this startup is dependent on the installation directory of <b>perl(1)</b> , which is not necessarily identical on all systems where WA2L/edrc is installed.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>lib/perl/pm/</b>           | additional Perl modules used by the Perl scripts started thru the <b>perl_wrapper(3)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>lib/ploticus/</b>          | files read by the ploticus command <b>pl(1)</b> , especially when using the <b>-prefab</b> option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>lib/python/</b>            | Python scripts called via <b>python_wrapper(1)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>lib/python/pym/</b>                                                                             | additional Python modules used by the Python scripts started thru the <b>python_wrapper(1)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>man/</b>                                                                                        | manual pages. See <b>edrcman(1)</b> and <b>manpages(4)</b> for an explanation of the organization of the manual pages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>scripts/</b>                                                                                    | here are all recovery scripts stored by default. The content of this directory has to be customized to your environment. For a description how to write recovery scripts for EDRC see <b>edrcscripts(1m)</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>scripts/sys/</b>                                                                                | Pre-configured "Recovery" script tree:<br>System Configuration handling scripts, including some automated EDRC tasks as a simple EDRC patch installation. The EDRC configuration file <b>edrc.sys.cfg</b> points to this directory.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>scripts/unknown-edrc/, scripts/unknown-sat/, scripts/unknown-sat/sup:Support/*sup:*Support/</b> | Dummy "Recovery" script trees that are shown when invoking <b>edrc</b> , <b>sat</b> , <b>asup</b> , <b>osup</b> and <b>psup</b> when the <i>Customer</i> in the <b>server_environment.cfg</b> configuration file cannot be resolved. Therefore, if the <b>edrc</b> , <b>sat</b> , <b>asup</b> , <b>osup</b> or <b>psup</b> commands load one of those trees, you need to set the entries in the <b>server_environment.cfg</b> properly.                                                                                                                                                                                                                                                    |
| <b>scripts/&lt;Customer&gt;-edrc/</b>                                                              | Disaster Recovery script tree to be created for disaster recovery for the <i>&lt;Customer&gt;</i> where the system belongs to. The <i>&lt;Customer&gt;</i> has to be set identical to the value returned by the <b>server_environment -C</b> command.<br>The default EDRC configuration file <b>edrc.cfg</b> points to this directory and is used if you invoke the <b>edrc</b> command without options.                                                                                                                                                                                                                                                                                   |
| <b>scripts/&lt;Customer&gt;-sat/</b>                                                               | Script tree to be created for System Administration to automate common tasks for the <i>&lt;Customer&gt;</i> where the system belongs to. The <i>&lt;Customer&gt;</i> has to be set identical to the value returned by the <b>server_environment -C</b> command.<br>The default EDRC configuration file <b>edrc.sat.cfg</b> points to this directory and is used if you invoke the <b>sat</b> command.                                                                                                                                                                                                                                                                                     |
| <b>scripts/&lt;Customer&gt;-sat/sup:Support/asup:ApplicationSupport/</b>                           | Script tree to be created for Application Support to automate common application operation support tasks for the <i>&lt;Customer&gt;</i> where the system belongs to. The <i>&lt;Customer&gt;</i> has to be set identical to the value returned by the <b>server_environment -C</b> command.<br>The EDRC configuration file <b>edrc.asup.cfg</b> points to this directory and is used if you invoke the <b>asup</b> command.<br><br>The <b>edrc.asup.cfg</b> configures less permissions (editing of menus and menu points etc.) for its users. Because it is a sub-directory of the <b>sat</b> menu tree, it can nevertheless be managed centrally by the user of the <b>sat</b> command. |

**scripts/<Customer>-sat/sup:Support/osup:OperationSupport/**

Script tree to be created for Operation Support to automate common operation support tasks for the <Customer> where the system belongs to. The <Customer> has to be set identical to the value returned by the **server\_environment -C** command.

The EDRC configuration file **edrc.osup.cfg** points to this directory and is used if you invoke the **asup** command.

The **edrc.osup.cfg** configures less permissions (editing of menus and menu points etc.) for its users. Because it is a sub-directory of the **sat** menu tree, it can nevertheless be managed centrally by the user of the **sat** command.

**scripts/<Customer>-sat/sup:Support/psup:ProductionSupport/**

Script tree to be created for Production Support to automate common production support tasks for the <Customer> where the system belongs to. The <Customer> has to be set identical to the value returned by the **server\_environment -C** command.

The EDRC configuration file **edrc.psup.cfg** points to this directory and is used if you invoke the **asup** command.

The **edrc.psup.cfg** configures less permissions (editing of menus and menu points etc.) for its users. Because it is a sub-directory of the **sat** menu tree, it can nevertheless be managed centrally by the user of the **sat** command.

**.sav**

backup directory in a recovery script tree. Whenever a file in **edrc** is edited or removed a backup copy is automatically saved into the **.sav** directory of the current menu (=subdir). Therefore, if you like to restore a file to a prior version, invoke the **shell** command in **edrc**, change to the **.sav** directory and restore the desired file using the normal operating system commands.

**src/**

source code of certain programs.

**var/**

var directory of WA2L/edrc.

**var/backup/**

several backups.

**var/barbedwire/**

**bwcreate**(1m) cache information.

**var/cache/**

cache base directory.

**var/cache/<command>/**

cache directory for a certain <command>.

**var/connection/**

this directory holds connection and security information used for connections to other hosts. See **edrc.cfg**(4) for more information about the currently possible *DIST\_MODE* settings and **edrc**(1m), **remote\_copy**(3) and **remote\_shell**(3) for a more detailed description of the **var/connection** directory.

|                                    |                                                                                                                                                                                                                                                                         |
|------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>var/contrib/edrc/</b>           | var directory for contributed commands to <b>edrc</b> . See <b>edrc(1m)</b> for information how to use this directory.                                                                                                                                                  |
| <b>var/csv2worksheet/</b>          | template spreadsheet files for the <b>csv2worksheet(1m)</b> command.                                                                                                                                                                                                    |
| <b>var/dns/&lt;dns-server&gt;/</b> | var directory of DNS server(s) bundled with the WA2L/edrc package.                                                                                                                                                                                                      |
| <b>var/edrcinit/</b>               | persistent data of <b>edrcinit(1m)</b> .                                                                                                                                                                                                                                |
| <b>var/fonts/</b>                  | additional fonts.                                                                                                                                                                                                                                                       |
| <b>var/locate/</b>                 | databases for the <b>locate(1)</b> command.                                                                                                                                                                                                                             |
| <b>var/lock/</b>                   | lockfiles. Do not edit them by hand.                                                                                                                                                                                                                                    |
| <b>var/log/</b>                    | all logfiles of the EDRC environment. Here the logfile of <b>edrc</b> and the output of all recovery scripts is saved. Therefore customizing scripts don't have to implement an own logging mechanism. All output sent to stdout will be saved in a designated logfile. |
| <b>var/logcheckd/</b>              | collect, report, state and pattern information used and written by the <b>logcheckd(1m)</b> command.                                                                                                                                                                    |
| <b>var/lots/</b>                   | default var directory of <b>lots(1m)</b> .                                                                                                                                                                                                                              |
| <b>var/makeuser/classes/</b>       | this directory contains example user classes files used by <b>makeuser</b> .                                                                                                                                                                                            |
| <b>var/manvi/</b>                  | manual page templates for <b>manvi(1)</b> .                                                                                                                                                                                                                             |
| <b>var/notes/</b>                  | free text notes. The contents of this directory is excluded from the package.                                                                                                                                                                                           |
| <b>var/omniutil/pd/</b>            | this directory contains the OBII Oracle pointer directory structure templates.                                                                                                                                                                                          |
| <b>var/pack/</b>                   | state information for <b>pack</b> . Do not edit it by hand.                                                                                                                                                                                                             |
| <b>var/passwdsyncd/</b>            | default spool directory for the password sync daemon <b>passwdsyncd</b> .                                                                                                                                                                                               |
| <b>var/pscount/</b>                | process reference information used by the <b>pscount(3)</b> command.                                                                                                                                                                                                    |
| <b>var/repl/</b>                   | this directory is used to save scripts replicated with the <b>distribute</b> EDRC command.                                                                                                                                                                              |
| <b>var/samples/</b>                | this directory contains configuration samples for reference and inspiration.                                                                                                                                                                                            |
| <b>var/settings/</b>               | persistent settings.                                                                                                                                                                                                                                                    |

|                                            |                                                                                                                                                         |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>var/shell/</b>                          | here the shell history of the interactive shell built in to <b>edrc</b> and the history of <b>edrc</b> are kept.                                        |
| <b>var/spool/</b>                          | spool directories for <b>crond</b> (3).                                                                                                                 |
| <b>var/sw/</b>                             | directory to <b>pack</b> (1m) the WA2L/edrc package to or to store delivered WA2L/edrc patches. The content of this directory is excluded from packing. |
| <b>var/sysconfig/development.acme.com/</b> | in this directory a template directory structure used for <b>sysconfig</b> is saved.                                                                    |
| <b>var/tmp/</b>                            | temporary files to be kept within the application.                                                                                                      |
| <b>var/trash/</b>                          | default 'Recycle Bin' directory for <b>trash</b> and the related commands.                                                                              |
| <b>var/vsav/</b>                           | directory to save versions of a file by <b>vsav</b> and related commands to it.                                                                         |
| <b>var/www/</b>                            | web files for <b>thttpd</b> (3) web server.                                                                                                             |

## SEE ALSO

**EDRC**(1), **ansi2txt**(3), **appendpdf**(3), **apply2file**(1), **apply2sw\_inventory**(1), **apprevious**(3), **app-root**(3), **apropos**(1), **ascii**(3), **asup**(1), **awk**(1), **banner**(3), **bash**(1), **batteryalert**(1), **bc**(1), **binprobe**(1m), **binprobe.dat**(4), **bget**(1), **bunzip2**(1), **busy**(1), **bwcreate**(1m), **bwcreate.cfg**(4), **bzcat**(1), **bzdiff**(1), **bzgrep**(1), **bzip2**(1), **bzip2recover**(1), **bzmore**(1), **catcomp**(1), **ccrypt**(1), **cfig2html**(1m), **checkopt**(3), **checkopt.h**(3), **choice**(3), **cltrash**(1), **cmdlist**(1m), **cmmon**(1m), **cmswitch**(1m), **comm**(1), **compatibility**(1), **connect**(3), **consolidate**(1), **contrib**(1m), **contrib.<contributed\_command>**(1m), **contrib.doc**(1m), **contrib.edrc**(1m), **contrib.pwsafe**(1m), **cpanm**(1), **cpio**(1), **crond**(3), **cronhandler**(1), **cronhandler.cfg**(4), **crontab**(4), **crypt**(1), **csv**(3), **csvcat**(3), **csvq**(3), **csv2worksheet**(3), **daemon\_wrapper**(1), **datalist.dat**(4), **days**(3), **DBI**(3), **dbrep**(1), **dbrep.cfg**(4), **diff**(1), **dig**(1), **directories**(3), **dmidecode**(3), **dos2ux**(3), **du\_index**(4), **du\_report**(1), **duvi**(1), **ecronnext**(1), **ecrontab**(1), **edrc**(1m), **edrc.cfg**(4), **edrcenv**(1), **edrc.help**(4), **edrcinit**(1m), **edrcinit.cfg**(4), **edrcinit.handler**(3), **edrclicense**(4), **edrcman**(1), **edrcman.cfg**(4), **edrcman.map**(4), **edrcpack**(1m), **edrcperm.no\_shell**(3), **edrcports**(4), **edrcrevision**(1), **edrcrevision.cfg**(4), **edrcroot**(3), **edrcscripts**(1m), **edrcsetup**(1m), **egrep**(1), **env**(4), **envpasswdstrip**(1), **epub2pdf**(1), **eshell**(1), **eterm**(1), **exiftool**(1), **expect**(3), **exrc**(4), **fcreate**(3), **fields2swvi**(3), **filedist**(1), **filedist.block**(4), **filedist.cfg**(4), **filegrep**(1), **filelink**(1), **filesize**(3), **filewatch**(3), **fit**(3), **fit2width**(3), **fnmatch**(3), **fmatch**(3), **freespace**(3), **fssum**(1), **fssum.cfg**(4), **ftps**(1), **gawk**(1), **gecos**(3), **getfilesystem**(3), **gethostbyname**(1), **getmountpoint**(3), **glibc.version**(3), **gid**(3), **grep**(1), **group**(3), **gs**(3), **gunzip**(1), **gzip**(1), **h2**(1), **histlist**(3), **homedir**(3), **hostaliases**(3), **hostlist**(3), **hostlist.cfg**(4), **hostlistdat2cfg**(3), **html2mht**(3), **hwinventory**(1), **ident**(3), **indent**(3), **info**(1), **ini.cfg**(4), **input**(3), **input\_targets**(3), **ipcalc**(1), **ipsort**(1), **is\_config\_byhand**(3), **is\_existing**(3), **is\_osid**(3), **is\_permitted**(3), **is\_running**(3), **is\_user**(3), **is\_up**(3), **is\_weekend**(3), **is\_writeable**(3), **java**(1), **java\_wrapper**(1), **java\_wrapper.cfg**(4), **job**(1), **joblog**(1), **jobstart**(1), **jobstart.cfg**(4), **jobstart.lib**(4), **jobwatch**(1), **jq**(3), **kalc**(1), **ksh**(1), **kshell**(3), **kshrc**(4), **ksh\_wrapper**(1), **ksh\_wrapper.cfg**(4), **lbanner**(3), **leo**(3), **lgcheckd**(1m), **lgcheckd.cfg**(4), **lgcheckd.state.db**(4), **lgcpattern**(3), **lgrep**(1), **libexpect**(3), **listtemp**(3), **ll**(1), **llcomp**(1), **locate**(1), **locatedb**(4), **locations**(1), **locations.cfg**(4), **log**(3), **logcheckd**(1m), **logcheckd.cfg**(4), **logcheckd.interface**(3), **logcheckd.pattern**(4), **logcheckd.style**(4), **logcut**(1), **logcut.cfg**(1), **log\_files.cfg**(4), **loggrep**(1), **logrotate**(1), **logrotate.cfg**(4), **logtail**(1), **logview**(1), **lots**(1m), **lots.cfg**(4), **lotsctl**(1), **lotsctl.cfg**(4), **lscmp**(1), **lscp**(1), **lsmv**(1), **lsof**(1), **lspm**(1), **lua**(1), **luac**(1), **luaenv**(3), **luarocks**(1), **luarocks.cfg**(4), **luaversion**(3), **lua\_wrapper**(1), **lua\_wrapper.cfg**(4), **lynx**(1),



lynx.cfg(4), lynx.lss(4), mail\_file(1), mail\_file.cfg(4), maketemp(3), makeuser(1), makeuser.cfg(4), man2html(3), mandir(4), manpages(4), manvi(1), manvi.cfg(4), mc(1), mcedit(1), mcview(1), md2html(3), md5string(3), mediawiki(3), mkuser(3), msg(3), msmt(3), name(1), nano(1), nanorc(4), nc(3), nginx(3), nginx.cfg(4), nginx.doc.cfg(4), nginx.mime\_types.cfg(4), nmap(1), nologin(1), nping(1), omniutil(1m), omniutil.cfg(4), oratab(4), osid(3), osid.cfg(4), osid.dat(4), osid.probe(3), osup(1), os\_wrapper(1), pack(1m), pack.cfg(4), passwdcombine(1), passwdsort(1), passwdsyncd(1m), passwdsyncd\_apply(3), passwdsyncd.cfg(4), patchinstall(1m), pdfscissors(1), perlenv(3), perlpod(4), perl\_modules(3), perlversion(3), perl\_wrapper(1), perl\_wrapper.cfg(4), pf\_wrapper(1), pip(1), pip3(1), pkgdir(1m), pkg\_hostname(3), pkg\_hostname.cfg(4), pkzip(1), pl(1), pmdesc(1), pod2html(3), portscan(3), print\_header(3), print\_index(3), print\_list(3), program.h(3), pscout(3), pscout.kp(4), psjoin(3), pslist(3), ps2pdf(3), pstree(1), psup(1), purgetemp(3), pwcrypt(1), pwsafe(3), python\_wrapper(1), python\_wrapper.cfg(4), pythonenv(3), pythonversion(3), random(3), rcac(1), rcomm(1), rcmd(1), rcmd.cfg(4), rdif(1), readline(3), regexintro(4), rel2abs(3), remote\_copy(3), remote\_copy.cfg(4), remote\_shell(3), remote\_shell.cfg(4), removetemp(3), repeat(3), resolve\_targetlist(3), revision(1), rl(1), rlogin(1), rnano(1), role\_option(3), role\_option.cfg(4), rosid(3), rsat(1), rsat.cfg(4), rssh(1), rsync(1), rsyncd.conf(4), sat(1), sav(1), schedule.dat(4), screen(1), scriptextract(3), scriptgrep(1), scriptheadersync(1), scriptmenupath(3), scriptrevision(3), scriptsequence(3), scripttitle(3), seconds(3), sectioncat(3), sed1line(1), select\_columns(3), server\_environment(3), server\_environment.cfg(4), sh(1), shell(1), shell.cfg(4), shellinabox(3), shlib(3), sortc(3), sortv(3), sparse(3), splitvt(1), sqlite(1), ssh-exec(1), ssh-exec.cfg(4), ssh-keyadd(1m), ssh-keyadd.cfg(4), ssh-keyadd.pub(4), sshpass(1), stat(1), strace(1), streamcat(3), strings.h(3), svi(1), sw\_inventory(4), sw\_report(1), swvi(1), sys(1), sysconfig(1), sysconfig.cfg(4), syscp(1), sysdiff(1), sysinfo(1), syspoll(1), syspoll.cfg(4), sysvi(1), sys2html(1m), sys2html.dat(4), sys2html.inf(4), tcpdump(1), termcap(4), textblock(3), textcolor(3), thttpd(3), thttpd.report.cfg(4), thttpd.doc.cfg(4), thttpd.foswiki.cfg(4), thttpd.report.cfg(4), timer(1), timezone(3), timezone.cfg(4), timezone.dat(4), title(1), tnsnames.ora(4), today(3), tolower(3), tomorrow(3), top(1), toupper(3), tpl(1), tpl.cfg(4), tput\_examples(3), traceroute(1), traceroute(1m), trash(1), trash.cfg(4), truss(1), tscat(3), tsize(3), tty\_columns(3), ttyplay(1), ttyrec(1), ttytime(1), tty\_variable(3), tzdump(3), uid(3), uncbz(1), uncbz2(1), uncz(1), unczst(1), unde(1), unipath(3), unrpm(1), untbz(1), untbz2(1), untgz(1), untrash(1), untzst(1), unzip(1), unzstd(1), updatedb(1m), usage(1), user(3), userclass.index(4), user\_info(1), user\_info.cfg(4), utility.h(3), uudecode(1), uuencode(1), uuencode(4), ux2dos(3), vadsp(1), vcat(1), vdiff(1), vgrep(1), vlist(1), vls(1), vmore(1), volume.dat(4), vpurge(1), vrestore(1), vsav(1), vsdfml(1), vvi(1), wa2lc(3), wa2ledrc\_edrcapi(1), wa2ledrc\_report(1), wa2ledrc\_shellinabxd(1), wa2l\_util(3), watchdog(1), watchdog.cfg(4), wget(1), whatis(1), whereami(1), whoisin(1), whoisin.list(4), winexe(1), wmic(1), woist(1), xbd(1), xlog(1), xmore(1), xml2csv(3), xml2json(3), xpid(1), xtee(1), xz(1), yesterday(3), ypxfr\_all(1), zcat(1), zdiff(1), zgrep(1), zip(1), zless(1), zmore(1), zstd(1)

## NOTES

Many thanks to Stefan Huber (EDS, Information Business GmbH, Switzerland). The OmniBack recovery scripts used in the disaster recovery project in 2003 were based on his excellent knowledge of HP-DataProtector (OmniBack).

Special thanks to Reimund Mueller (EDS, Information Business GmbH, Switzerland), who acted as editor of the "1 / Allgemein" part of the "Emergency Handbook" in his leisure time.

The WA2L/edrc package came into being during a disaster recovery project realized for an insurance company in Switzerland in 2003. A huge portion of the WA2L/edrc package has been developed by the author Christian Walther <wa2l@users.sourceforge.net> in his leisure time.

Now in 2023 the WA2L/edrc has a lifespan of 20 years; it turns out that it is still useful for new disaster recovery solutions and system administration simplification.

The source of the commands **sav(1)**, **homedir(3)**, **mkuser(3)**, **woist(1)** and **pid(1)** is the **SFI-Director**.

The following commands were inspired by the **SFI-Director**: **osid(3)**, **nologin(1)** and **sysconfig(1)**. The **SFI-Director**, was an excellent (GPL) toolset for managing NIS and NIS++, distributing software over several Unix/Linux flavors, central user profile management, configuring and documenting systems and much more developed by Peter Stevens. Unlike many other script based tools all commands were realized as very reliable and stable shell scripts and demonstrated by example that it is possible to implement reliable and portable applications as shell scripts. All commands and files of the **SFI Director** were well documented. The sources of the script based **SFI-Director** are no longer available online.

Check out other WA2L projects on Sourceforge: <https://sourceforge.net/u/wa2l/profile/>.

## LICENSE

Copyright © 2003 - 2025 Christian Walther

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

A copy of the GNU General Public License is distributed with WA2L/edrc and is saved in **edrc/doc/COPYING**.

Check on Sourceforge (<http://sourceforge.net/projects/wa2l-edrc/>) for the most recent version of the WA2L/edrc package.

## BUGS

Bugs and limitations are generally documented in the related manual pages.

## AUTHOR

WA2L/edrc was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**NAME**

ansi2txt – Translates vt100 codes into readable text (or html)

**SYNOPSIS**

**ansi2txt** [OPTION] [INPUT FILE]

**DESCRIPTION**

Ansi2txt converts a stream of vt100 (ansi) codes on the stdin into readable text on the stdout. An INPUT file can be specified instead of stdin.

Ansi2html (ansi2txt -html) renders the vt100 codes into a html file that looks quite a lot like your terminal did.

**OPTIONS****-w WIDTH**

If specified the output rendering screen will be limited to WIDTH characters wide.

**-h HEIGHT**

If specified the output rendering screen will be limited to HEIGHT character rows, overflow will overprint.

**-rv** Reverse video

**-html** Output as html

**-txt** Output as text

**-refresh secs**

In html mode add <meta http-equiv="refresh" content="secs"> to <head>

**-v** Version

**--help** Displays usage information

**BUGS**

Double height text in html mode only renders the top row, the bottom double height text row is not rendered. Double width is rendered as ordinary characters separated by spaces. Scrolling windows are not implemented.

**NAME**

appendpdf – Append one PDF to another

**SYNOPSIS**

```
appendpdf [options] file1.pdf file2.pdf [outfile.pdf]
```

**Options:**

|              |                                               |
|--------------|-----------------------------------------------|
| -p --prepend | prepend the document instead of appending it  |
| -f --forms   | wipe all forms and annotations from the PDF   |
| -o --order   | preserve the internal PDF ordering for output |
| -v --verbose | print diagnostic messages                     |
| -h --help    | verbose help message                          |
| -V --version | print CAM::PDF version                        |

**DESCRIPTION**

Copy the contents of file2.pdf to the end of file1.pdf. This may break complex PDFs which include forms, so the --forms option is provided to eliminate those elements from the resulting PDF.

**SEE ALSO**

CAM::PDF

*deletepdfpage.pl*

**AUTHOR**

See CAM::PDF

**NAME**

apply2file – apply data stream to file

**SYNOPSIS**

**edrc/bin/apply2file** [ **-h** ]

**apply2file** [ **-o** ] [ **-s** ] [ **-u** *user* ] [ **-g** *group* ] [ **-p** *perm* ] **-f** *target\_file*

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

Apply a data stream provided via pipe to a target file. The target file is untouched if the data stream does not differ from the target file. This is useful to edit files or to generate files which are system dependent.

**OPTIONS**

- h**           usage message.
- o**           if target file exists, preserve target file permissions. In this case the options **-u**, **-g** and **-p** are ignored.
- s**           save the target file prior to the applying of the data stream. To save the target file the **sav(1)** command is used internally.
- u** *user*     username or UID of the target file.
- g** *group*    groupname or GID of the target file.
- p** *perm*     filepermissions of the target file.
- f** *target\_file*  
              target file where to install the stream.

**EXIT STATUS**

- 0**           no error.
- 2**           operating system not supported, see **osid(3)**
- 4**           usage displayed.

- 5 the data is identical with the existing file and the data stream is not applied to the target file.
- 6 **apply2file** could not write to the target file.

## FILES

-

## EXAMPLES

- 1) write system information to /etc/issue

```
{
 echo
 host=`hostname`
 banner $host
 echo
 uname -rsm
 echo
} | \
apply2file -o -u root -g root -p 640 \
-f /etc/issue
```

- 2) enable the startup of a daemon (samba in this case)

```
cat /etc/rc.config.d/samba | \
sed -e 's/RUN_SAMBA=.* /RUN_SAMBA=1/g' | \
apply2file -o -f /etc/rc.config.d/samba
```

## SEE ALSO

**edrcintro(1)**, **sav(1)**, **chmod(1)**, **chown(1)**

## NOTES

**apply2file** should be used in **pre\_exec** and **post\_exec** scripts of **sysconfig** to avoid altering of system files if not needed.

## BUGS

-

**AUTHOR**

apply2file was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

apply2sw\_inventory – apply data stream with named swvi fields to software inventory file

**SYNOPSIS**

edrc/bin/apply2sw\_inventory [ **-h** ]

apply2sw\_inventory [ **-d** *output\_dir* ] [ **-r** ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

This is the scripting frontend to the interactive **swvi** command.

With **apply2sw\_inventory** it is possible to automatically update a software inventory file in a install script without the need to enter the software inventory information by hand using the **swvi** command or to write directly to the **..sw\_inventory** file which format might change in future releases.

For more information about the software inventory creation and maintenance see **swvi**(1) and **sw\_report**(1).

As in the **swvi** command, only the entries for **SOFTWARE**, **VERSION**, **INST\_USER** and **INST\_GROUP** are compulsory. The sequence of the entries provided is not significant. For a complete list and description of all entries see **swvi**(1).

See section **EXAMPLES** for common usages of **apply2sw\_inventory** in an install script.

**OPTIONS**

**-h**           usage message.

**-d** *output\_dir*  
directory where the software inventory is modified.

**-r**           remove (clear) contents of an eventually existing inventory file. This option equals to the **-r** option of **swvi**. If not specified the software specification provided is appended (equals to **swvi -a**) to the software inventory file.

**ENVIRONMENT**

-



**EXIT STATUS**

|          |                                                               |
|----------|---------------------------------------------------------------|
| <b>0</b> | no error.                                                     |
| <b>1</b> | software inventory not modified.                              |
| <b>2</b> | operating system not supported, see <b>osid(3)</b> .          |
| <b>3</b> | cannot write to the <i>output_dir</i> directory.              |
| <b>4</b> | usage displayed.                                              |
| <b>5</b> | the command has been aborted pressing <i>Ctrl+C</i> .         |
| <b>6</b> | <b>apply2sw_inventory</b> could not write to the target file. |

**FILES****..sw\_inventory**

software inventory file. This file contains the software information (version, name, ...) for a software installed into a certain directory and is used to create a software inventory with the **sw\_report(1)** command.

**EXAMPLES**

- 1) Install script for automated installation of WA2L/edrc having an interactive install procedure

This are the major parts of a simplified installation script to give an idea of updating/creating the software inventory file without manual intervention.

This installation script installs WA2L/edrc that is available in the package file **/dat/sw/apps/edrc-1.5.02/edrc-1.5.02-20070428.sh** to the installation directory **/opt**. All manual input queried by the shell archive **edrc-1.5.02-20070428.sh** is provided via input redirection. After that the software inventory file is updated using **apply2sw\_inventory**. Because in this case WA2L/edrc is the only software in the installation directory the probably existing software inventory file is cleared using the **-r** option ahead of writing to it.

```
#!/bin/sh
#
My_INSTALL - My Install Script for: EDRC Shell archive
#
[00] 21.09.2005 CWa Initial Version
#

Const
PATH=$PATH:~edrc/bin ; export PATH
Installbasedir=/opt
Pkgfile=/dat/sw/apps/edrc-1.5.02/edrc-1.5.02-20070428.sh

package installation
```

```

#
cat <<EOM | $Pkgfile
n
$Installbasedir
n
n
y
y
EOM

update software inventory
#
cat <<EOM | apply2sw_inventory -r -d $Installbasedir/edrc
 SOFTWARE = EDRC - Enterprise Disaster Recovery Console
 VERSION = 1.5.02
 INST_USER = root
 INST_GROUP = root
 RUN_USER =
 RUN_GROUP =
 CFG_DIR = etc
 VAR_DIR = var
 LICENSE = GNU GPL
 LIC_MGMT =
 WEB =
 DOC = ~edrc/doc/edrc_manpages-1.5.02.pdf
 INST_SRC = $Pkgfile
 PRODUCT = System
 INSTANCE =
 COMMENT = disaster recovery and system administration
EOM

```

## 2) Install script of the GNU C compilers using a HP-UX depot

This are the major parts of a simplified installation script to give an idea of updating/creating the software inventory file without manual intervention.

This installation script installs the GNU C compilers that are available in the HP-UX depot file **/dat/sw/apps/gcc-3.3.2/gcc-3.3.2-hppa-11.11.depot.gz** to the installation directory **/usr/local**. The installation is performed with the **swinstall** command of HP-UX. After that the software inventory file is updated using **apply2sw\_inventory**. Due to the fact that in **/usr/local** often many different software are installed the most likely existing software inventory should not be cleared ahead of writing to it. That's why **apply2sw\_inventory** is called without the remove (clear) option.

```

#!/bin/sh
#
My_INSTALL - My Install Script for: GNU C Compiler
#
[00] 21.09.2005 CWa Initial Version
#

Const
PATH=$PATH:~edrc/bin ; export PATH
Installbasedir=/usr/local

```

```

Pkgfile=/dat/sw/apps/gcc-3.3.2/gcc-3.3.2-hppa-11.11.depot.gz

package installation
#
cd /
swinstall -s $Pkgfile gcc

update software inventory
#
cat <<EOM | apply2sw_inventory -d $Installbasedir
 SOFTWARE = GNU C Compilers
 VERSION = 3.3.2
 INST_USER = root
 INST_GROUP = sys
 RUN_USER =
 RUN_GROUP =
 CFG_DIR =
 VAR_DIR =
 LICENSE = GNU GPL
 LIC_MGMT =
 WEB = http://hpux.cs.utah.edu/hppd/hpux/Gnu/gcc-3.3.2
 DOC = http://gcc.gnu.org
 INST_SRC = $Pkgfile
 PRODUCT = Development
 INSTANCE =
 COMMENT =
EOM

```

**SEE ALSO**

**edrcintro(1), sw\_inventory(4) sw\_report(1), swvi(1), fields2swvi(3)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

apply2sw\_inventory was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

apply2sw\_inventory(1)

General Commands

apply2sw\_inventory(1)

## **COPYRIGHT**

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## NAME

approot – print the root of the WA2L/edrc installation

## SYNOPSIS

**edrc/lib/approot**

## AVAILABILITY

WA2L/edrc

## DESCRIPTION

With **approot** the root of the WA2L/edrc software installation is printed. This command is used inside the command to evaluate the basedir.

## EXIT STATUS

**0** always

## SEE ALSO

**edrcintro(1)**, **edrcroot(3)**

## NOTES

**approot** will replace **edrcroot** completely as soon as all commands has been migrated to use **approot** instead of **edrcroot**.

## BUGS

-

## AUTHOR

approot was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

## COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

## NAME

apprevision – return (version) revision number of WA2L/edrc

## SYNOPSIS

**edrc/lib/apprevision**

## AVAILABILITY

WA2L/edrc

## DESCRIPTION

The **apprevision** program returns the revision number of the WA2L/edrc package.

The output is identical to **sat -V** and **edrc -V**, but the response is much faster. The exit codes and the error messages of **apprevision** is identical to the calls of **sat -V** and **edrc -V**.

## OPTIONS

-

## ENVIRONMENT

-

## EXIT STATUS

**3** version printed.

**9** the **edrcrevision.cfg** file is missing or the **VERSION** or **PATCHLEVEL** options within this file are not set properly.

## FILES

**etc/edrcrevision.cfg**

configuration file which holds the current version and patchlevel of EDRC. See **edrcrevision.cfg(4)** for more information.

## EXAMPLES

-

## SEE ALSO

**edrc**(1m), **edrcintro**(1), **edrcrevision.cfg**(4), **patchinstall**(1m), **sat**(1m)

## NOTES

-

## BUGS

-

## AUTHOR

apprevision was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

## COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

apropos – search the whatis database for strings

**SYNOPSIS**

**edrc/bin/apropos** *keyword* ...

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

**apropos** searches a set of database files containing short descriptions of system commands for keywords and displays the result on the standard output.

**OPTIONS**

**keyword** keyword to be searched.

**FILES**

**whatis** whatis databases in the \$MANPATH.

**SEE ALSO**

**edrcintro(1)**, **edrcman(1)**, **man(1)**, **whatis(1)**

**NOTES**

Parts of this manpage were extracted from the documentation of the apropos man page written by John W. Eaton and modified to fit into WA2L/edrc package.

**BUGS**

-

**AUTHOR**

apropos was developed by John W. Eaton, <jwe@che.utexas.edu>, Department of Chemical Engineering, The University of Texas at Austin, Austin Texas 78712 and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

**COPYRIGHT**

Copyright © 2008 Christian Walther



This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

ascii – ASCII character set encoded in octal, decimal, and hexadecimal

**DESCRIPTION**

ASCII is the American Standard Code for Information Interchange. It is a 7-bit code. Many 8-bit codes (e.g., ISO 8859-1) contain ASCII as their lower half. The international counterpart of ASCII is known as ISO 646-IRV.

The following table contains the 128 ASCII characters.

C program '\x' escapes are noted.

| Oct | Dec | Hex | Char                      | Oct | Dec | Hex | Char   |
|-----|-----|-----|---------------------------|-----|-----|-----|--------|
| 000 | 0   | 00  | NUL '\0' (null character) | 100 | 64  | 40  | @      |
| 001 | 1   | 01  | SOH (start of heading)    | 101 | 65  | 41  | A      |
| 002 | 2   | 02  | STX (start of text)       | 102 | 66  | 42  | B      |
| 003 | 3   | 03  | ETX (end of text)         | 103 | 67  | 43  | C      |
| 004 | 4   | 04  | EOT (end of transmission) | 104 | 68  | 44  | D      |
| 005 | 5   | 05  | ENQ (enquiry)             | 105 | 69  | 45  | E      |
| 006 | 6   | 06  | ACK (acknowledge)         | 106 | 70  | 46  | F      |
| 007 | 7   | 07  | BEL '\a' (bell)           | 107 | 71  | 47  | G      |
| 010 | 8   | 08  | BS '\b' (backspace)       | 110 | 72  | 48  | H      |
| 011 | 9   | 09  | HT '\t' (horizontal tab)  | 111 | 73  | 49  | I      |
| 012 | 10  | 0A  | LF '\n' (new line)        | 112 | 74  | 4A  | J      |
| 013 | 11  | 0B  | VT '\v' (vertical tab)    | 113 | 75  | 4B  | K      |
| 014 | 12  | 0C  | FF '\f' (form feed)       | 114 | 76  | 4C  | L      |
| 015 | 13  | 0D  | CR '\r' (carriage ret)    | 115 | 77  | 4D  | M      |
| 016 | 14  | 0E  | SO (shift out)            | 116 | 78  | 4E  | N      |
| 017 | 15  | 0F  | SI (shift in)             | 117 | 79  | 4F  | O      |
| 020 | 16  | 10  | DLE (data link escape)    | 120 | 80  | 50  | P      |
| 021 | 17  | 11  | DC1 (device control 1)    | 121 | 81  | 51  | Q      |
| 022 | 18  | 12  | DC2 (device control 2)    | 122 | 82  | 52  | R      |
| 023 | 19  | 13  | DC3 (device control 3)    | 123 | 83  | 53  | S      |
| 024 | 20  | 14  | DC4 (device control 4)    | 124 | 84  | 54  | T      |
| 025 | 21  | 15  | NAK (negative ack.)       | 125 | 85  | 55  | U      |
| 026 | 22  | 16  | SYN (synchronous idle)    | 126 | 86  | 56  | V      |
| 027 | 23  | 17  | ETB (end of trans. blk)   | 127 | 87  | 57  | W      |
| 030 | 24  | 18  | CAN (cancel)              | 130 | 88  | 58  | X      |
| 031 | 25  | 19  | EM (end of medium)        | 131 | 89  | 59  | Y      |
| 032 | 26  | 1A  | SUB (substitute)          | 132 | 90  | 5A  | Z      |
| 033 | 27  | 1B  | ESC (escape)              | 133 | 91  | 5B  | [      |
| 034 | 28  | 1C  | FS (file separator)       | 134 | 92  | 5C  | \ '\\' |
| 035 | 29  | 1D  | GS (group separator)      | 135 | 93  | 5D  | ]      |
| 036 | 30  | 1E  | RS (record separator)     | 136 | 94  | 5E  | ^      |
| 037 | 31  | 1F  | US (unit separator)       | 137 | 95  | 5F  | _      |
| 040 | 32  | 20  | SPACE                     | 140 | 96  | 60  | `      |
| 041 | 33  | 21  | !                         | 141 | 97  | 61  | a      |
| 042 | 34  | 22  | "                         | 142 | 98  | 62  | b      |
| 043 | 35  | 23  | #                         | 143 | 99  | 63  | c      |
| 044 | 36  | 24  | \$                        | 144 | 100 | 64  | d      |
| 045 | 37  | 25  | %                         | 145 | 101 | 65  | e      |
| 046 | 38  | 26  | &                         | 146 | 102 | 66  | f      |
| 047 | 39  | 27  | '                         | 147 | 103 | 67  | g      |
| 050 | 40  | 28  | (                         | 150 | 104 | 68  | h      |
| 051 | 41  | 29  | )                         | 151 | 105 | 69  | i      |
| 052 | 42  | 2A  | *                         | 152 | 106 | 6A  | j      |

|     |    |    |   |     |     |    |     |
|-----|----|----|---|-----|-----|----|-----|
| 053 | 43 | 2B | + | 153 | 107 | 6B | k   |
| 054 | 44 | 2C | , | 154 | 108 | 6C | l   |
| 055 | 45 | 2D | - | 155 | 109 | 6D | m   |
| 056 | 46 | 2E | . | 156 | 110 | 6E | n   |
| 057 | 47 | 2F | / | 157 | 111 | 6F | o   |
| 060 | 48 | 30 | 0 | 160 | 112 | 70 | p   |
| 061 | 49 | 31 | 1 | 161 | 113 | 71 | q   |
| 062 | 50 | 32 | 2 | 162 | 114 | 72 | r   |
| 063 | 51 | 33 | 3 | 163 | 115 | 73 | s   |
| 064 | 52 | 34 | 4 | 164 | 116 | 74 | t   |
| 065 | 53 | 35 | 5 | 165 | 117 | 75 | u   |
| 066 | 54 | 36 | 6 | 166 | 118 | 76 | v   |
| 067 | 55 | 37 | 7 | 167 | 119 | 77 | w   |
| 070 | 56 | 38 | 8 | 170 | 120 | 78 | x   |
| 071 | 57 | 39 | 9 | 171 | 121 | 79 | y   |
| 072 | 58 | 3A | : | 172 | 122 | 7A | z   |
| 073 | 59 | 3B | ; | 173 | 123 | 7B | {   |
| 074 | 60 | 3C | < | 174 | 124 | 7C |     |
| 075 | 61 | 3D | = | 175 | 125 | 7D | }   |
| 076 | 62 | 3E | > | 176 | 126 | 7E | ~   |
| 077 | 63 | 3F | ? | 177 | 127 | 7F | DEL |

## Tables

For convenience, below are more compact tables in hex and decimal.

| 2 3 4 5 6 7      | 30 40 50 60 70 80 90 100 110 120 |
|------------------|----------------------------------|
| 0: 0 @ P ` p     | 0: ( 2 < F P Z d n x             |
| 1: ! 1 A Q a q   | 1: ) 3 = G Q [ e o y             |
| 2: " 2 B R b r   | 2: * 4 > H R \ f p z             |
| 3: # 3 C S c s   | 3: ! + 5 ? I S ] g q {           |
| 4: \$ 4 D T d t  | 4: " , 6 @ J T ^ h r             |
| 5: % 5 E U e u   | 5: # - 7 A K U _ i s }           |
| 6: & 6 F V f v   | 6: \$ . 8 B L V ` j t ~          |
| 7: ' 7 G W g w   | 7: % / 9 C M W a k u DEL         |
| 8: ( 8 H X h x   | 8: & 0 : D N X b l v             |
| 9: ) 9 I Y i y   | 9: ' 1 ; E O Y c m w             |
| A: * : J Z j z   |                                  |
| B: + ; K [ k {   |                                  |
| C: , < L \ l     |                                  |
| D: - = M ] m }   |                                  |
| E: . > N ^ n ~   |                                  |
| F: / ? O _ o DEL |                                  |

## NOTES

### History

An **ascii** manual page appeared in Version 7 of AT&T UNIX.

On older terminals, the underscore code is displayed as a left arrow, called backarrow, the caret is displayed as an up-arrow and the vertical bar has a hole in the middle.

Uppercase and lowercase characters differ by just one bit and the ASCII character 2 differs from the double quote by just one bit, too. That made it much easier to encode characters mechanically or with a non-microcontroller-based electronic keyboard and that pairing was found on old teletypes.

The ASCII standard was published by the United States of America Standards Institute (USASI) in 1968.

**SEE ALSO**

**charsets(3), utf-8(3)**

**COLOPHON**

This page is part of release 4.15 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.

**NAME**

asup – start edrc with a special Application Support configuration

**SYNOPSIS**

**edrc/bin/asup** [ **-h** | **-V** ]

**asup** [ **-s** ] [ **-t** ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

Short start of **edrc** with an other configuration which points to an own script tree. Internally **asup** calls **edrc -c edrc.asup.cfg -n ASUP\_@ID@** .

**asup** stands for "Application SUPport".

The session name (as shown in the menu) is automatically set to **ASUP\_<id>** . Where **<id>** is the process id of the started **edrc** instance if not already a session with the same name exists, if so the **<id>** is set to a random number.

If additional short starts are needed, create a symlink from the new short start command to **sat** ( see section **EXAMPLES** ).

**OPTIONS**

**-h**           usage message.

**-V**           print version and patch level of **edrc**. For an explanation of the release numbering system see **edrcrevision(1)**.

**-s**           silent startup. Startup without showing the EDRC banner.

**-t**           no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

**ENVIRONMENT**

-

**EXIT STATUS**

see **edrc**(1m).

**FILES**

**etc/edrc.asup.cfg**

configuration file of **asup**, see **edrc.cfg**(4) for more information.

Other files see section **FILES** in **edrc**(1m).

**EXAMPLES****1) create a new short start**

This creates a new short start command **new\_shortstart** that will load the configuration file **edrc.new\_shortstart.cfg**. The session name will automatically be set to **NEW\_SHORTSTART\_<id>** .

```
[/root]
[root@rh7mzv7t001] [bash]: cd ~edrc/bin

[/opt/edrc/bin]
[root@rh7mzv7t001] [bash]: ln -s sat new_shortstart
```

**SEE ALSO**

**sat**(1), **edrc**(1m), **edrc.cfg**(4), **edrcintro**(1), **edrcrevision**(1)

**NOTES**

-

**BUGS**

-

**AUTHOR**

**asup** was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net) .

**COPYRIGHT**

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

awk – pattern-directed scanning and processing language

**SYNOPSIS**

**awk** [ **-F** *fs* ] [ **-v** *var=value* ] [ '*prog*' | **-f** *progfile* ] [ *file* ... ]

**DESCRIPTION**

*Awk* scans each input *file* for lines that match any of a set of patterns specified literally in *prog* or in one or more files specified as **-f** *progfile*. With each pattern there can be an associated action that will be performed when a line of a *file* matches the pattern. Each line is matched against the pattern portion of every pattern-action statement; the associated action is performed for each matched pattern. The file name **-** means the standard input. Any *file* of the form *var=value* is treated as an assignment, not a filename, and is executed at the time it would have been opened if it were a filename. The option **-v** followed by *var=value* is an assignment to be done before *prog* is executed; any number of **-v** options may be present. The **-F** *fs* option defines the input field separator to be the regular expression *fs*.

An input line is normally made up of fields separated by white space, or by regular expression **FS**. The fields are denoted **\$1**, **\$2**, ..., while **\$0** refers to the entire line. If **FS** is null, the input line is split into one field per character.

A pattern-action statement has the form

```
pattern { action }
```

A missing { *action* } means print the line; a missing pattern always matches. Pattern-action statements are separated by newlines or semicolons.

An action is a sequence of statements. A statement can be one of the following:

```
if(expression) statement [else statement]
while(expression) statement
for(expression ; expression ; expression) statement
for(var in array) statement
do statement while(expression)
break
continue
{ [statement ...] }
expression # commonly var = expression
print [expression-list] [> expression]
printf format [, expression-list] [> expression]
return [expression]
next # skip remaining patterns on this input line
nextfile # skip rest of this file, open next, start at top
delete array[expression] # delete an array element
delete array # delete all elements of array
exit [expression] # exit immediately; status is expression
```

Statements are terminated by semicolons, newlines or right braces. An empty *expression-list* stands for **\$0**. String constants are quoted " ", with the usual C escapes recognized within. Expressions take on string or numeric values as appropriate, and are built using the operators + - \* / % ^ (exponentiation), and concatenation (indicated by white space). The operators ! ++ -- += -= \*= /= %= ^= > >= < <= == != ?: are also available in expressions. Variables may be scalars, array elements (denoted *x[i]*) or fields. Variables are initialized to the null string. Array subscripts may be any string, not necessarily numeric; this allows for a form of associative memory. Multiple subscripts such as [*i,j,k*] are permitted; the constituents are concatenated, separated by the value of **SUBSEP**.

The **print** statement prints its arguments on the standard output (or on a file if **>file** or **>>file** is present or on a pipe if **|cmd** is present), separated by the current output field separator, and terminated by the output record separator. *file* and *cmd* may be literal names or parenthesized expressions; identical string values in different statements denote the same open file. The **printf** statement formats its expression list according to



the format (see *printf*(3)). The built-in function **close**(*expr*) closes the file or pipe *expr*. The built-in function **fflush**(*expr*) flushes any buffered output for the file or pipe *expr*.

The mathematical functions **exp**, **log**, **sqrt**, **sin**, **cos**, and **atan2** are built in. Other built-in functions:

**length** the length of its argument taken as a string, or of **\$0** if no argument.

**rand** random number on (0,1)

**srand** sets seed for **rand** and returns the previous seed.

**int** truncates to an integer value

**substr**(*s, m, n*)

the *n*-character substring of *s* that begins at position *m* counted from 1.

**index**(*s, t*)

the position in *s* where the string *t* occurs, or 0 if it does not.

**match**(*s, r*)

the position in *s* where the regular expression *r* occurs, or 0 if it does not. The variables **RSTART** and **RLENGTH** are set to the position and length of the matched string.

**split**(*s, a, fs*)

splits the string *s* into array elements *a*[1], *a*[2], ..., *a*[*n*], and returns *n*. The separation is done with the regular expression *fs* or with the field separator **FS** if *fs* is not given. An empty string as field separator splits the string into one array element per character.

**sub**(*r, t, s*)

substitutes *t* for the first occurrence of the regular expression *r* in the string *s*. If *s* is not given, **\$0** is used.

**gsub** same as **sub** except that all occurrences of the regular expression are replaced; **sub** and **gsub** return the number of replacements.

**sprintf**(*fmt, expr, ...*)

the string resulting from formatting *expr ...* according to the *printf*(3) format *fmt*

**system**(*cmd*)

executes *cmd* and returns its exit status

**tolower**(*str*)

returns a copy of *str* with all upper-case characters translated to their corresponding lower-case equivalents.

**toupper**(*str*)

returns a copy of *str* with all lower-case characters translated to their corresponding upper-case equivalents.

The “function” **getline** sets **\$0** to the next input record from the current input file; **getline <file** sets **\$0** to the next record from *file*. **getline x** sets variable *x* instead. Finally, *cmd* | **getline** pipes the output of *cmd* into **getline**; each call of **getline** returns the next line of output from *cmd*. In all cases, **getline** returns 1 for a successful input, 0 for end of file, and -1 for an error.

Patterns are arbitrary Boolean combinations (with **!**, **||**, and **&&**) of regular expressions and relational expressions. Regular expressions are as in *egrep*; see *grep*(1). Isolated regular expressions in a pattern apply to the entire line. Regular expressions may also occur in relational expressions, using the operators **~** and **!~**. */rel* is a constant regular expression; any string (constant or variable) may be used as a regular expression, except in the position of an isolated regular expression in a pattern.

A pattern may consist of two patterns separated by a comma; in this case, the action is performed for all lines from an occurrence of the first pattern though an occurrence of the second.

A relational expression is one of the following:

*expression matchop regular-expression*  
*expression relop expression*  
*expression in array-name*  
*(expr,expr,...) in array-name*

where a relop is any of the six relational operators in C, and a matchop is either ~ (matches) or !~ (does not match). A conditional is an arithmetic expression, a relational expression, or a Boolean combination of these.

The special patterns **BEGIN** and **END** may be used to capture control before the first input line is read and after the last. **BEGIN** and **END** do not combine with other patterns.

Variable names with special meanings:

#### **CONVFMT**

conversion format used when converting numbers (default **%.6g**)

**FS** regular expression used to separate fields; also settable by option **-Ffs**.

**NF** number of fields in the current record

**NR** ordinal number of the current record

**FNR** ordinal number of the current record in the current file

#### **FILENAME**

the name of the current input file

**RS** input record separator (default newline)

**OFS** output field separator (default blank)

**ORS** output record separator (default newline)

**OFMT** output format for numbers (default **%.6g**)

#### **SUBSEP**

separates multiple subscripts (default 034)

**ARGC** argument count, assignable

**ARGV** argument array, assignable; non-null members are taken as filenames

#### **ENVIRON**

array of environment variables; subscripts are names.

Functions may be defined (at the position of a pattern-action statement) thus:

**function foo(a, b, c) { ...; return x }**

Parameters are passed by value if scalar and by reference if array name; functions may be called recursively. Parameters are local to the function; all other variables are global. Thus local variables may be created by providing excess parameters in the function definition.

#### **EXAMPLES**

```
length($0) > 72
 Print lines longer than 72 characters.

{ print $2, $1 }
 Print first two fields in opposite order.

BEGIN { FS = ", [\t]* | [\t]+" }
 { print $2, $1 }
 Same, with input fields separated by comma and/or blanks and tabs.

 { s += $1 }

END { print "sum is", s, " average is", s/NR }
 Add up first column, print sum and average.
```

```
/start/, /stop/
```

Print all lines between start/stop pairs.

```
BEGIN { # Simulate echo(1)
 for (i = 1; i < ARGV; i++) printf "%s ", ARGV[i]
 printf "\n"
 exit }
```

## SEE ALSO

*lex*(1), *sed*(1)

A. V. Aho, B. W. Kernighan, P. J. Weinberger, *The AWK Programming Language*, Addison-Wesley, 1988.  
ISBN 0-201-07981-X

## BUGS

There are no explicit conversions between numbers and strings. To force an expression to be treated as a number add 0 to it; to force it to be treated as a string concatenate "" to it.

The scope rules for variables in functions are a botch; the syntax is worse.

**NAME**

banner – make posters in large letters

**SYNOPSIS**

**edrc/lib/banner** "*strings*"

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

**banner** prints its arguments (each with a maximum of 10 characters) in large letters to **stdout** .

Each argument is printed on a separate line. Note that multiple-word strings must be enclosed in quotes in order to be printed on the same line.

**OPTIONS**

usage message.

"*a string*" string to be printed in large letters. The qotes are only needed when a multiple word string has to be printed on the same output line.

**ENVIRONMENT**

-

**EXIT STATUS**

**0** no error.

**4** usage displayed.

**FILES**

-

**EXAMPLES**

-

**SEE ALSO**

**edrcintro**(1), **lbanner**(3)

**NOTES**

The **banner** command was compiled from the source code **banner.c** provided by Mark S. Kolich on <http://mark.koli.ch/2008/11/howto-use-the-banner-banner-c-command-to-create-login-banners.html> and has been integrated into WA2L/edrc by Christian Walther.

**BUGS**

-

**AUTHOR**

banner was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

**bash** – GNU Bourne-Again SHell

**SYNOPSIS**

**bash** [options] [file]

**COPYRIGHT**

Bash is Copyright © 1989-2005 by the Free Software Foundation, Inc.

**DESCRIPTION**

**Bash** is an **sh**-compatible command language interpreter that executes commands read from the standard input or from a file. **Bash** also incorporates useful features from the *Korn* and *C* shells (**ksh** and **csh**).

**Bash** is intended to be a conformant implementation of the Shell and Utilities portion of the IEEE POSIX specification (IEEE Standard 1003.1). **Bash** can be configured to be POSIX-conformant by default.

**OPTIONS**

In addition to the single-character shell options documented in the description of the **set** builtin command, **bash** interprets the following options when it is invoked:

- c** *string* If the **-c** option is present, then commands are read from *string*. If there are arguments after the *string*, they are assigned to the positional parameters, starting with **\$0**.
- i** If the **-i** option is present, the shell is *interactive*.
- l** Make **bash** act as if it had been invoked as a login shell (see **INVOCATION** below).
- r** If the **-r** option is present, the shell becomes *restricted* (see **RESTRICTED SHELL** below).
- s** If the **-s** option is present, or if no arguments remain after option processing, then commands are read from the standard input. This option allows the positional parameters to be set when invoking an interactive shell.
- v** Print shell input lines as they are read.
- x** Print commands and their arguments as they are executed.
- D** A list of all double-quoted strings preceded by **\$** is printed on the standard output. These are the strings that are subject to language translation when the current locale is not **C** or **POSIX**. This implies the **-n** option; no commands will be executed.
- [+O]** *[shopt\_option]*  
*shopt\_option* is one of the shell options accepted by the **shopt** builtin (see **SHELL BUILTIN COMMANDS** below). If *shopt\_option* is present, **-O** sets the value of that option; **+O** unsets it. If *shopt\_option* is not supplied, the names and values of the shell options accepted by **shopt** are printed on the standard output. If the invocation option is **+O**, the output is displayed in a format that may be reused as input.
- A **--** signals the end of options and disables further option processing. Any arguments after the **--** are treated as filenames and arguments. An argument of **-** is equivalent to **--**.

**Bash** also interprets a number of multi-character options. These options must appear on the command line before the single-character options to be recognized.

**--debugger**

Arrange for the debugger profile to be executed before the shell starts. Turns on extended debugging mode (see the description of the **extdebug** option to the **shopt** builtin below) and shell function tracing (see the description of the **-o functrace** option to the **set** builtin below).

**--dump-po-strings**

Equivalent to **-D**, but the output is in the GNU *gettext* **po** (portable object) file format.

**--dump-strings**

Equivalent to **-D**.

**--help** Display a usage message on standard output and exit successfully.

**--init-file** *file***--rcfile** *file*

Execute commands from *file* instead of the standard personal initialization file *~/.bashrc* if the shell is interactive (see **INVOCATION** below).

**--login**

Equivalent to **-l**.

**--noediting**

Do not use the GNU **readline** library to read command lines when the shell is interactive.

**--noprofile**

Do not read either the system-wide startup file */etc/profile* or any of the personal initialization files *~/.bash\_profile*, *~/.bash\_login*, or *~/.profile*. By default, **bash** reads these files when it is invoked as a login shell (see **INVOCATION** below).

**--norc**

Do not read and execute the personal initialization file *~/.bashrc* if the shell is interactive. This option is on by default if the shell is invoked as **sh**.

**--posix**

Change the behavior of **bash** where the default operation differs from the POSIX standard to match the standard (*posix mode*).

**--restricted**

The shell becomes restricted (see **RESTRICTED SHELL** below).

**--verbose**

Equivalent to **-v**.

**--version**

Show version information for this instance of **bash** on the standard output and exit successfully.

**ARGUMENTS**

If arguments remain after option processing, and neither the **-c** nor the **-s** option has been supplied, the first argument is assumed to be the name of a file containing shell commands. If **bash** is invoked in this fashion, **\$0** is set to the name of the file, and the positional parameters are set to the remaining arguments. **Bash** reads and executes commands from this file, then exits. **Bash**'s exit status is the exit status of the last command executed in the script. If no commands are executed, the exit status is 0. An attempt is first made to open the file in the current directory, and, if no file is found, then the shell searches the directories in **PATH** for the script.

**INVOCATION**

A *login shell* is one whose first character of argument zero is a **-**, or one started with the **--login** option.

An *interactive* shell is one started without non-option arguments and without the **-c** option whose standard input and error are both connected to terminals (as determined by *isatty(3)*), or one started with the **-i** option. **PS1** is set and **\$-** includes **i** if **bash** is interactive, allowing a shell script or a startup file to test this state.

The following paragraphs describe how **bash** executes its startup files. If any of the files exist but cannot be read, **bash** reports an error. Tildes are expanded in file names as described below under **Tilde Expansion** in the **EXPANSION** section.

When **bash** is invoked as an interactive login shell, or as a non-interactive shell with the **--login** option, it first reads and executes commands from the file */etc/profile*, if that file exists. After reading that file, it looks for *~/.bash\_profile*, *~/.bash\_login*, and *~/.profile*, in that order, and reads and executes commands from the first one that exists and is readable. The **--noprofile** option may be used when the shell is started to inhibit this behavior.

When a login shell exits, **bash** reads and executes commands from the file *~/.bash\_logout*, if it exists.

When an interactive shell that is not a login shell is started, **bash** reads and executes commands from *~/.bashrc*, if that file exists. This may be inhibited by using the **--norc** option. The **--rcfile file** option will force **bash** to read and execute commands from *file* instead of *~/.bashrc*.

When **bash** is started non-interactively, to run a shell script, for example, it looks for the variable **BASH\_ENV** in the environment, expands its value if it appears there, and uses the expanded value as the name of a file to read and execute. **Bash** behaves as if the following command were executed:

```
if [-n "$BASH_ENV"]; then . "$BASH_ENV"; fi
```

but the value of the **PATH** variable is not used to search for the file name.

If **bash** is invoked with the name **sh**, it tries to mimic the startup behavior of historical versions of **sh** as closely as possible, while conforming to the POSIX standard as well. When invoked as an interactive login shell, or a non-interactive shell with the **--login** option, it first attempts to read and execute commands from */etc/profile* and *~/.profile*, in that order. The **--noprofile** option may be used to inhibit this behavior. When invoked as an interactive shell with the name **sh**, **bash** looks for the variable **ENV**, expands its value if it is defined, and uses the expanded value as the name of a file to read and execute. Since a shell invoked as **sh** does not attempt to read and execute commands from any other startup files, the **--rcfile** option has no effect. A non-interactive shell invoked with the name **sh** does not attempt to read any other startup files. When invoked as **sh**, **bash** enters *posix* mode after the startup files are read.

When **bash** is started in *posix* mode, as with the **--posix** command line option, it follows the POSIX standard for startup files. In this mode, interactive shells expand the **ENV** variable and commands are read and executed from the file whose name is the expanded value. No other startup files are read.

**Bash** attempts to determine when it is being run by the remote shell daemon, usually *rshd*. If **bash** determines it is being run by *rshd*, it reads and executes commands from *~/.bashrc*, if that file exists and is readable. It will not do this if invoked as **sh**. The **--norc** option may be used to inhibit this behavior, and the **--rcfile** option may be used to force another file to be read, but *rshd* does not generally invoke the shell with those options or allow them to be specified.

If the shell is started with the effective user (group) id not equal to the real user (group) id, and the **-p** option is not supplied, no startup files are read, shell functions are not inherited from the environment, the **SHELLOPTS** variable, if it appears in the environment, is ignored, and the effective user id is set to the real user id. If the **-p** option is supplied at invocation, the startup behavior is the same, but the effective user id is not reset.

## DEFINITIONS

The following definitions are used throughout the rest of this document.

**blank** A space or tab.

**word** A sequence of characters considered as a single unit by the shell. Also known as a **token**.

**name** A *word* consisting only of alphanumeric characters and underscores, and beginning with an alphabetic character or an underscore. Also referred to as an **identifier**.

**metacharacter**

A character that, when unquoted, separates words. One of the following:

| & ; ( ) < > space tab

**control operator**

A *token* that performs a control function. It is one of the following symbols:

|| & && ; ; ( ) | <newline>

## RESERVED WORDS

*Reserved words* are words that have a special meaning to the shell. The following words are recognized as reserved when unquoted and either the first word of a simple command (see **SHELL GRAMMAR** below) or the third word of a **case** or **for** command:

! case do done elif else esac fi for function if in select then until  
while { } time [[ ]]

## SHELL GRAMMAR

### Simple Commands

A *simple command* is a sequence of optional variable assignments followed by **blank**-separated words and redirections, and terminated by a *control operator*. The first word specifies the command to be executed, and is passed as argument zero. The remaining words are passed as arguments to the invoked command.

The return value of a *simple command* is its exit status, or 128+*n* if the command is terminated by signal *n*.



## Pipelines

A *pipeline* is a sequence of one or more commands separated by the character `|`. The format for a pipeline is:

```
[time [-p]] [!] command [| command2 ...]
```

The standard output of *command* is connected via a pipe to the standard input of *command2*. This connection is performed before any redirections specified by the command (see **REDIRECTION** below).

The return status of a pipeline is the exit status of the last command, unless the **pipefail** option is enabled. If **pipefail** is enabled, the pipeline's return status is the value of the last (rightmost) command to exit with a non-zero status, or zero if all commands exit successfully. If the reserved word **!** precedes a pipeline, the exit status of that pipeline is the logical negation of the exit status as described above. The shell waits for all commands in the pipeline to terminate before returning a value.

If the **time** reserved word precedes a pipeline, the elapsed as well as user and system time consumed by its execution are reported when the pipeline terminates. The **-p** option changes the output format to that specified by POSIX. The **TIMEFORMAT** variable may be set to a format string that specifies how the timing information should be displayed; see the description of **TIMEFORMAT** under **Shell Variables** below.

Each command in a pipeline is executed as a separate process (i.e., in a subshell).

## Lists

A *list* is a sequence of one or more pipelines separated by one of the operators **;**, **&**, **&&**, or **||**, and optionally terminated by one of **;**, **&**, or **<newline>**.

Of these list operators, **&&** and **||** have equal precedence, followed by **;** and **&**, which have equal precedence.

A sequence of one or more newlines may appear in a *list* instead of a semicolon to delimit commands.

If a command is terminated by the control operator **&**, the shell executes the command in the *background* in a subshell. The shell does not wait for the command to finish, and the return status is 0. Commands separated by a **;** are executed sequentially; the shell waits for each command to terminate in turn. The return status is the exit status of the last command executed.

The control operators **&&** and **||** denote AND lists and OR lists, respectively. An AND list has the form

```
command1 && command2
```

*command2* is executed if, and only if, *command1* returns an exit status of zero.

An OR list has the form

```
command1 || command2
```

*command2* is executed if and only if *command1* returns a non-zero exit status. The return status of AND and OR lists is the exit status of the last command executed in the list.

## Compound Commands

A *compound command* is one of the following:

(*list*) *list* is executed in a subshell environment (see **COMMAND EXECUTION ENVIRONMENT** below). Variable assignments and builtin commands that affect the shell's environment do not remain in effect after the command completes. The return status is the exit status of *list*.

{ *list*; } *list* is simply executed in the current shell environment. *list* must be terminated with a newline or semicolon. This is known as a *group command*. The return status is the exit status of *list*. Note that unlike the metacharacters ( and ), { and } are *reserved words* and must occur where a reserved word is permitted to be recognized. Since they do not cause a word break, they must be separated from *list* by whitespace.

((*expression*))

The *expression* is evaluated according to the rules described below under **ARITHMETIC EVALUATION**. If the value of the expression is non-zero, the return status is 0; otherwise the return status is 1. This is exactly equivalent to **let "expression"**.

**[[ *expression* ]]**

Return a status of 0 or 1 depending on the evaluation of the conditional expression *expression*. Expressions are composed of the primaries described below under **CONDITIONAL EXPRESSIONS**. Word splitting and pathname expansion are not performed on the words between the **[[** and **]]**; tilde expansion, parameter and variable expansion, arithmetic expansion, command substitution, process substitution, and quote removal are performed. Conditional operators such as **-f** must be unquoted to be recognized as primaries.

When the **==** and **!=** operators are used, the string to the right of the operator is considered a pattern and matched according to the rules described below under **Pattern Matching**. If the shell option **nocasematch** is enabled, the match is performed without regard to the case of alphabetic characters. The return value is 0 if the string matches (**==**) or does not match (**!=**) the pattern, and 1 otherwise. Any part of the pattern may be quoted to force it to be matched as a string.

An additional binary operator, **=~**, is available, with the same precedence as **==** and **!=**. When it is used, the string to the right of the operator is considered an extended regular expression and matched accordingly (as in *regex(3)*). The return value is 0 if the string matches the pattern, and 1 otherwise. If the regular expression is syntactically incorrect, the conditional expression's return value is 2. If the shell option **nocasematch** is enabled, the match is performed without regard to the case of alphabetic characters. Substrings matched by parenthesized subexpressions within the regular expression are saved in the array variable **BASH\_REMATCH**. The element of **BASH\_REMATCH** with index 0 is the portion of the string matching the entire regular expression. The element of **BASH\_REMATCH** with index *n* is the portion of the string matching the *n*th parenthesized subexpression.

Expressions may be combined using the following operators, listed in decreasing order of precedence:

**( *expression* )**

Returns the value of *expression*. This may be used to override the normal precedence of operators.

**! *expression***

True if *expression* is false.

***expression1* && *expression2***

True if both *expression1* and *expression2* are true.

***expression1* || *expression2***

True if either *expression1* or *expression2* is true.

The **&&** and **||** operators do not evaluate *expression2* if the value of *expression1* is sufficient to determine the return value of the entire conditional expression.

**for *name* [ **in** *word* ] ; **do** *list* ; **done****

The list of words following **in** is expanded, generating a list of items. The variable *name* is set to each element of this list in turn, and *list* is executed each time. If the **in** *word* is omitted, the **for** command executes *list* once for each positional parameter that is set (see **PARAMETERS** below). The return status is the exit status of the last command that executes. If the expansion of the items following **in** results in an empty list, no commands are executed, and the return status is 0.

**for (( *expr1* ; *expr2* ; *expr3* )) ; **do** *list* ; **done****

First, the arithmetic expression *expr1* is evaluated according to the rules described below under **ARITHMETIC EVALUATION**. The arithmetic expression *expr2* is then evaluated repeatedly until it evaluates to zero. Each time *expr2* evaluates to a non-zero value, *list* is executed and the arithmetic expression *expr3* is evaluated. If any expression is omitted, it behaves as if it evaluates to 1. The return value is the exit status of the last command in *list* that is executed, or false if any of the expressions is invalid.

**select *name* [ **in** *word* ] ; **do** *list* ; **done****

The list of words following **in** is expanded, generating a list of items. The set of expanded words is printed on the standard error, each preceded by a number. If the **in** *word* is omitted, the

positional parameters are printed (see **PARAMETERS** below). The **PS3** prompt is then displayed and a line read from the standard input. If the line consists of a number corresponding to one of the displayed words, then the value of *name* is set to that word. If the line is empty, the words and prompt are displayed again. If EOF is read, the command completes. Any other value read causes *name* to be set to null. The line read is saved in the variable **REPLY**. The *list* is executed after each selection until a **break** command is executed. The exit status of **select** is the exit status of the last command executed in *list*, or zero if no commands were executed.

**case word in** [ (*pattern* [*pattern*] ... ) *list* ;; ] ... **esac**

A **case** command first expands *word*, and tries to match it against each *pattern* in turn, using the same matching rules as for pathname expansion (see **Pathname Expansion** below). The *word* is expanded using tilde expansion, parameter and variable expansion, arithmetic substitution, command substitution, process substitution and quote removal. Each *pattern* examined is expanded using tilde expansion, parameter and variable expansion, arithmetic substitution, command substitution, and process substitution. If the shell option **nocasematch** is enabled, the match is performed without regard to the case of alphabetic characters. When a match is found, the corresponding *list* is executed. After the first match, no subsequent matches are attempted. The exit status is zero if no pattern matches. Otherwise, it is the exit status of the last command executed in *list*.

**if list; then list; [ elif list; then list; ] ... [ else list; ] fi**

The **if list** is executed. If its exit status is zero, the **then list** is executed. Otherwise, each **elif list** is executed in turn, and if its exit status is zero, the corresponding **then list** is executed and the command completes. Otherwise, the **else list** is executed, if present. The exit status is the exit status of the last command executed, or zero if no condition tested true.

**while list; do list; done**

**until list; do list; done**

The **while** command continuously executes the **do list** as long as the last command in *list* returns an exit status of zero. The **until** command is identical to the **while** command, except that the test is negated; the **do list** is executed as long as the last command in *list* returns a non-zero exit status. The exit status of the **while** and **until** commands is the exit status of the last **do list** command executed, or zero if none was executed.

### Shell Function Definitions

A shell function is an object that is called like a simple command and executes a compound command with a new set of positional parameters. Shell functions are declared as follows:

[ **function** ] *name* () *compound-command* [*redirection*]

This defines a function named *name*. The reserved word **function** is optional. If the **function** reserved word is supplied, the parentheses are optional. The *body* of the function is the compound command *compound-command* (see **Compound Commands** above). That command is usually a *list* of commands between { and }, but may be any command listed under **Compound Commands** above. *compound-command* is executed whenever *name* is specified as the name of a simple command. Any redirections (see **REDIRECTION** below) specified when a function is defined are performed when the function is executed. The exit status of a function definition is zero unless a syntax error occurs or a readonly function with the same name already exists. When executed, the exit status of a function is the exit status of the last command executed in the body. (See **FUNCTIONS** below.)

### COMMENTS

In a non-interactive shell, or an interactive shell in which the **interactive\_comments** option to the **shopt** builtin is enabled (see **SHELL BUILTIN COMMANDS** below), a word beginning with # causes that word and all remaining characters on that line to be ignored. An interactive shell without the **interactive\_comments** option enabled does not allow comments. The **interactive\_comments** option is on by default in interactive shells.

## QUOTING

*Quoting* is used to remove the special meaning of certain characters or words to the shell. Quoting can be used to disable special treatment for special characters, to prevent reserved words from being recognized as such, and to prevent parameter expansion.

Each of the *metacharacters* listed above under **DEFINITIONS** has special meaning to the shell and must be quoted if it is to represent itself.

When the command history expansion facilities are being used (see **HISTORY EXPANSION** below), the *history expansion* character, usually `!`, must be quoted to prevent history expansion.

There are three quoting mechanisms: the *escape character*, single quotes, and double quotes.

A non-quoted backslash (`\`) is the *escape character*. It preserves the literal value of the next character that follows, with the exception of `<newline>`. If a `\<newline>` pair appears, and the backslash is not itself quoted, the `\<newline>` is treated as a line continuation (that is, it is removed from the input stream and effectively ignored).

Enclosing characters in single quotes preserves the literal value of each character within the quotes. A single quote may not occur between single quotes, even when preceded by a backslash.

Enclosing characters in double quotes preserves the literal value of all characters within the quotes, with the exception of `$`, `'`, `\`, and, when history expansion is enabled, `!`. The characters `$` and `'` retain their special meaning within double quotes. The backslash retains its special meaning only when followed by one of the following characters: `$`, `'`, `"`, `\`, or `<newline>`. A double quote may be quoted within double quotes by preceding it with a backslash. If enabled, history expansion will be performed unless an `!` appearing in double quotes is escaped using a backslash. The backslash preceding the `!` is not removed.

The special parameters `*` and `@` have special meaning when in double quotes (see **PARAMETERS** below).

Words of the form `$'string'` are treated specially. The word expands to *string*, with backslash-escaped characters replaced as specified by the ANSI C standard. Backslash escape sequences, if present, are decoded as follows:

|                   |                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------|
| <code>\a</code>   | alert (bell)                                                                                   |
| <code>\b</code>   | backspace                                                                                      |
| <code>\e</code>   | an escape character                                                                            |
| <code>\f</code>   | form feed                                                                                      |
| <code>\n</code>   | new line                                                                                       |
| <code>\r</code>   | carriage return                                                                                |
| <code>\t</code>   | horizontal tab                                                                                 |
| <code>\v</code>   | vertical tab                                                                                   |
| <code>\\</code>   | backslash                                                                                      |
| <code>\'</code>   | single quote                                                                                   |
| <code>\nnn</code> | the eight-bit character whose value is the octal value <i>nnn</i> (one to three digits)        |
| <code>\xHH</code> | the eight-bit character whose value is the hexadecimal value <i>HH</i> (one or two hex digits) |
| <code>\cx</code>  | a control- <i>x</i> character                                                                  |

The expanded result is single-quoted, as if the dollar sign had not been present.

A double-quoted string preceded by a dollar sign (`$`) will cause the string to be translated according to the current locale. If the current locale is **C** or **POSIX**, the dollar sign is ignored. If the string is translated and replaced, the replacement is double-quoted.

## PARAMETERS

A *parameter* is an entity that stores values. It can be a *name*, a number, or one of the special characters listed below under **Special Parameters**. A *variable* is a parameter denoted by a *name*. A variable has a *value* and zero or more *attributes*. Attributes are assigned using the **declare** builtin command (see **declare** below in **SHELL BUILTIN COMMANDS**).

A parameter is set if it has been assigned a value. The null string is a valid value. Once a variable is set, it may be unset only by using the **unset** builtin command (see **SHELL BUILTIN COMMANDS** below).

A *variable* may be assigned to by a statement of the form

```
name=[value]
```

If *value* is not given, the variable is assigned the null string. All *values* undergo tilde expansion, parameter and variable expansion, command substitution, arithmetic expansion, and quote removal (see **EXPANSION** below). If the variable has its **integer** attribute set, then *value* is evaluated as an arithmetic expression even if the  $\$(...)$  expansion is not used (see **Arithmetic Expansion** below). Word splitting is not performed, with the exception of "\$@" as explained below under **Special Parameters**. Pathname expansion is not performed. Assignment statements may also appear as arguments to the **alias**, **declare**, **typeset**, **export**, **readonly**, and **local** builtin commands.

In the context where an assignment statement is assigning a value to a shell variable or array index, the += operator can be used to append to or add to the variable's previous value. When += is applied to a variable for which the integer attribute has been set, *value* is evaluated as an arithmetic expression and added to the variable's current value, which is also evaluated. When += is applied to an array variable using compound assignment (see **Arrays** below), the variable's value is not unset (as it is when using =), and new values are appended to the array beginning at one greater than the array's maximum index. When applied to a string-valued variable, *value* is expanded and appended to the variable's value.

### Positional Parameters

A *positional parameter* is a parameter denoted by one or more digits, other than the single digit 0. Positional parameters are assigned from the shell's arguments when it is invoked, and may be reassigned using the **set** builtin command. Positional parameters may not be assigned to with assignment statements. The positional parameters are temporarily replaced when a shell function is executed (see **FUNCTIONS** below).

When a positional parameter consisting of more than a single digit is expanded, it must be enclosed in braces (see **EXPANSION** below).

### Special Parameters

The shell treats several parameters specially. These parameters may only be referenced; assignment to them is not allowed.

- \* Expands to the positional parameters, starting from one. When the expansion occurs within double quotes, it expands to a single word with the value of each parameter separated by the first character of the **IFS** special variable. That is, "\$\*" is equivalent to "\$1c\$2c...", where *c* is the first character of the value of the **IFS** variable. If **IFS** is unset, the parameters are separated by spaces. If **IFS** is null, the parameters are joined without intervening separators.
- @ Expands to the positional parameters, starting from one. When the expansion occurs within double quotes, each parameter expands to a separate word. That is, "\$@" is equivalent to "\$1" "\$2" ... If the double-quoted expansion occurs within a word, the expansion of the first parameter is joined with the beginning part of the original word, and the expansion of the last parameter is joined with the last part of the original word. When there are no positional parameters, "\$@" and @\$ expand to nothing (i.e., they are removed).
- # Expands to the number of positional parameters in decimal.
- ? Expands to the status of the most recently executed foreground pipeline.
- Expands to the current option flags as specified upon invocation, by the **set** builtin command, or those set by the shell itself (such as the **-i** option).
- \$ Expands to the process ID of the shell. In a () subshell, it expands to the process ID of the current shell, not the subshell.
- ! Expands to the process ID of the most recently executed background (asynchronous) command.
- 0 Expands to the name of the shell or shell script. This is set at shell initialization. If **bash** is invoked with a file of commands, \$0 is set to the name of that file. If **bash** is started with the **-c** option, then \$0 is set to the first argument after the string to be executed, if one is present. Otherwise, it is set to the file name used to invoke **bash**, as given by argument zero.
- At shell startup, set to the absolute pathname used to invoke the shell or shell script being executed as passed in the environment or argument list. Subsequently, expands to the last argument to the previous command, after expansion. Also set to the full pathname used to invoke each command executed and placed in the environment exported to that command. When checking mail, this

parameter holds the name of the mail file currently being checked.

### Shell Variables

The following variables are set by the shell:

**BASH** Expands to the full file name used to invoke this instance of **bash**.

#### **BASH\_ARGC**

An array variable whose values are the number of parameters in each frame of the current bash execution call stack. The number of parameters to the current subroutine (shell function or script executed with **.** or **source**) is at the top of the stack. When a subroutine is executed, the number of parameters passed is pushed onto **BASH\_ARGC**. The shell sets **BASH\_ARGC** only when in extended debugging mode (see the description of the **extdebug** option to the **shopt** builtin below)

#### **BASH\_ARGV**

An array variable containing all of the parameters in the current bash execution call stack. The final parameter of the last subroutine call is at the top of the stack; the first parameter of the initial call is at the bottom. When a subroutine is executed, the parameters supplied are pushed onto **BASH\_ARGV**. The shell sets **BASH\_ARGV** only when in extended debugging mode (see the description of the **extdebug** option to the **shopt** builtin below)

#### **BASH\_COMMAND**

The command currently being executed or about to be executed, unless the shell is executing a command as the result of a trap, in which case it is the command executing at the time of the trap.

#### **BASH\_EXECUTION\_STRING**

The command argument to the **-c** invocation option.

#### **BASH\_LINENO**

An array variable whose members are the line numbers in source files corresponding to each member of **FUNCNAME**. **\${BASH\_LINENO[\$i]}** is the line number in the source file where **\${FUNCNAME[\$i]}** was called. The corresponding source file name is **\${BASH\_SOURCE[\$i]}**. Use **LINENO** to obtain the current line number.

#### **BASH\_REMATCH**

An array variable whose members are assigned by the **=~** binary operator to the **[[** conditional command. The element with index 0 is the portion of the string matching the entire regular expression. The element with index *n* is the portion of the string matching the *n*th parenthesized subexpression. This variable is read-only.

#### **BASH\_SOURCE**

An array variable whose members are the source filenames corresponding to the elements in the **FUNCNAME** array variable.

#### **BASH\_SUBSHELL**

Incremented by one each time a subshell or subshell environment is spawned. The initial value is 0.

#### **BASH\_VERSINFO**

A readonly array variable whose members hold version information for this instance of **bash**. The values assigned to the array members are as follows:

|                         |                                                 |
|-------------------------|-------------------------------------------------|
| <b>BASH_VERSINFO[0]</b> | The major version number (the <i>release</i> ). |
| <b>BASH_VERSINFO[1]</b> | The minor version number (the <i>version</i> ). |
| <b>BASH_VERSINFO[2]</b> | The patch level.                                |
| <b>BASH_VERSINFO[3]</b> | The build version.                              |
| <b>BASH_VERSINFO[4]</b> | The release status (e.g., <i>beta1</i> ).       |
| <b>BASH_VERSINFO[5]</b> | The value of <b>MACHTYPE</b> .                  |

#### **BASH\_VERSION**

Expands to a string describing the version of this instance of **bash**.

#### **COMP\_CWORD**

An index into **\${COMP\_WORDS}** of the word containing the current cursor position. This variable is available only in shell functions invoked by the programmable completion facilities (see **Programmable Completion** below).

**COMP\_LINE**

The current command line. This variable is available only in shell functions and external commands invoked by the programmable completion facilities (see **Programmable Completion** below).

**COMP\_POINT**

The index of the current cursor position relative to the beginning of the current command. If the current cursor position is at the end of the current command, the value of this variable is equal to **\${#COMP\_LINE}**. This variable is available only in shell functions and external commands invoked by the programmable completion facilities (see **Programmable Completion** below).

**COMP\_WORDBREAKS**

The set of characters that the Readline library treats as word separators when performing word completion. If **COMP\_WORDBREAKS** is unset, it loses its special properties, even if it is subsequently reset.

**COMP\_WORDS**

An array variable (see **Arrays** below) consisting of the individual words in the current command line. The words are split on shell metacharacters as the shell parser would separate them. This variable is available only in shell functions invoked by the programmable completion facilities (see **Programmable Completion** below).

**DIRSTACK**

An array variable (see **Arrays** below) containing the current contents of the directory stack. Directories appear in the stack in the order they are displayed by the **dirs** builtin. Assigning to members of this array variable may be used to modify directories already in the stack, but the **pushd** and **popd** builtins must be used to add and remove directories. Assignment to this variable will not change the current directory. If **DIRSTACK** is unset, it loses its special properties, even if it is subsequently reset.

**EUID** Expands to the effective user ID of the current user, initialized at shell startup. This variable is readonly.

**FUNCNAME**

An array variable containing the names of all shell functions currently in the execution call stack. The element with index 0 is the name of any currently-executing shell function. The bottom-most element is "main". This variable exists only when a shell function is executing. Assignments to **FUNCNAME** have no effect and return an error status. If **FUNCNAME** is unset, it loses its special properties, even if it is subsequently reset.

**GROUPS**

An array variable containing the list of groups of which the current user is a member. Assignments to **GROUPS** have no effect and return an error status. If **GROUPS** is unset, it loses its special properties, even if it is subsequently reset.

**HISTCMD**

The history number, or index in the history list, of the current command. If **HISTCMD** is unset, it loses its special properties, even if it is subsequently reset.

**HOSTNAME**

Automatically set to the name of the current host.

**HOSTTYPE**

Automatically set to a string that uniquely describes the type of machine on which **bash** is executing. The default is system-dependent.

**LINENO**

Each time this parameter is referenced, the shell substitutes a decimal number representing the current sequential line number (starting with 1) within a script or function. When not in a script or function, the value substituted is not guaranteed to be meaningful. If **LINENO** is unset, it loses its special properties, even if it is subsequently reset.

**MACHTYPE**

Automatically set to a string that fully describes the system type on which **bash** is executing, in the standard GNU *cpu-company-system* format. The default is system-dependent.

**OLDPWD**

The previous working directory as set by the **cd** command.

**OPTARG**

The value of the last option argument processed by the **getopts** builtin command (see **SHELL BUILTIN COMMANDS** below).

**OPTIND**

The index of the next argument to be processed by the **getopts** builtin command (see **SHELL BUILTIN COMMANDS** below).

**OSTYPE**

Automatically set to a string that describes the operating system on which **bash** is executing. The default is system-dependent.

**PIPESTATUS**

An array variable (see **Arrays** below) containing a list of exit status values from the processes in the most-recently-executed foreground pipeline (which may contain only a single command).

**PPID** The process ID of the shell's parent. This variable is readonly.

**PWD** The current working directory as set by the **cd** command.

**RANDOM**

Each time this parameter is referenced, a random integer between 0 and 32767 is generated. The sequence of random numbers may be initialized by assigning a value to **RANDOM**. If **RANDOM** is unset, it loses its special properties, even if it is subsequently reset.

**REPLY**

Set to the line of input read by the **read** builtin command when no arguments are supplied.

**SECONDS**

Each time this parameter is referenced, the number of seconds since shell invocation is returned. If a value is assigned to **SECONDS**, the value returned upon subsequent references is the number of seconds since the assignment plus the value assigned. If **SECONDS** is unset, it loses its special properties, even if it is subsequently reset.

**SHELLOPTS**

A colon-separated list of enabled shell options. Each word in the list is a valid argument for the **-o** option to the **set** builtin command (see **SHELL BUILTIN COMMANDS** below). The options appearing in **SHELLOPTS** are those reported as *on* by **set -o**. If this variable is in the environment when **bash** starts up, each shell option in the list will be enabled before reading any startup files. This variable is read-only.

**SHLVL**

Incremented by one each time an instance of **bash** is started.

**UID** Expands to the user ID of the current user, initialized at shell startup. This variable is readonly.

The following variables are used by the shell. In some cases, **bash** assigns a default value to a variable; these cases are noted below.

**BASH\_ENV**

If this parameter is set when **bash** is executing a shell script, its value is interpreted as a filename containing commands to initialize the shell, as in *~/.bashrc*. The value of **BASH\_ENV** is subjected to parameter expansion, command substitution, and arithmetic expansion before being interpreted as a file name. **PATH** is not used to search for the resultant file name.



**CDPATH**

The search path for the **cd** command. This is a colon-separated list of directories in which the shell looks for destination directories specified by the **cd** command. A sample value is `". : ~ : /usr "`.

**COLUMNS**

Used by the **select** builtin command to determine the terminal width when printing selection lists. Automatically set upon receipt of a SIGWINCH.

**COMPREPLY**

An array variable from which **bash** reads the possible completions generated by a shell function invoked by the programmable completion facility (see **Programmable Completion** below).

**EMACS**

If **bash** finds this variable in the environment when the shell starts with value `t`, it assumes that the shell is running in an emacs shell buffer and disables line editing.

**FCEDIT**

The default editor for the **fc** builtin command.

**FIGNORE**

A colon-separated list of suffixes to ignore when performing filename completion (see **README** below). A filename whose suffix matches one of the entries in **FIGNORE** is excluded from the list of matched filenames. A sample value is `". o : ~ "`.

**GLOBIGNORE**

A colon-separated list of patterns defining the set of filenames to be ignored by pathname expansion. If a filename matched by a pathname expansion pattern also matches one of the patterns in **GLOBIGNORE**, it is removed from the list of matches.

**HISTCONTROL**

A colon-separated list of values controlling how commands are saved on the history list. If the list of values includes *ignorespace*, lines which begin with a **space** character are not saved in the history list. A value of *ignoredups* causes lines matching the previous history entry to not be saved. A value of *ignoreboth* is shorthand for *ignorespace* and *ignoredups*. A value of *erasedups* causes all previous lines matching the current line to be removed from the history list before that line is saved. Any value not in the above list is ignored. If **HISTCONTROL** is unset, or does not include a valid value, all lines read by the shell parser are saved on the history list, subject to the value of **HISTIGNORE**. The second and subsequent lines of a multi-line compound command are not tested, and are added to the history regardless of the value of **HISTCONTROL**.

**HISTFILE**

The name of the file in which command history is saved (see **HISTORY** below). The default value is `~/.bash_history`. If unset, the command history is not saved when an interactive shell exits.

**HISTFILESIZE**

The maximum number of lines contained in the history file. When this variable is assigned a value, the history file is truncated, if necessary, by removing the oldest entries, to contain no more than that number of lines. The default value is 500. The history file is also truncated to this size after writing it when an interactive shell exits.

**HISTIGNORE**

A colon-separated list of patterns used to decide which command lines should be saved on the history list. Each pattern is anchored at the beginning of the line and must match the complete line (no implicit `*` is appended). Each pattern is tested against the line after the checks specified by **HISTCONTROL** are applied. In addition to the normal shell pattern matching characters, `&` matches the previous history line. `&` may be escaped using a backslash; the backslash is removed before attempting a match. The second and subsequent lines of a multi-line compound command are not tested, and are added to the history regardless of the value of **HISTIGNORE**.

**HISTSIZE**

The number of commands to remember in the command history (see **HISTORY** below). The default value is 500.

**HISTTIMEFORMAT**

If this variable is set and not null, its value is used as a format string for *strftime*(3) to print the time stamp associated with each history entry displayed by the **history** builtin. If this variable is set, time stamps are written to the history file so they may be preserved across shell sessions.

**HOME**

The home directory of the current user; the default argument for the **cd** builtin command. The value of this variable is also used when performing tilde expansion.

**HOSTFILE**

Contains the name of a file in the same format as */etc/hosts* that should be read when the shell needs to complete a hostname. The list of possible hostname completions may be changed while the shell is running; the next time hostname completion is attempted after the value is changed, **bash** adds the contents of the new file to the existing list. If **HOSTFILE** is set, but has no value, **bash** attempts to read */etc/hosts* to obtain the list of possible hostname completions. When **HOSTFILE** is unset, the hostname list is cleared.

**IFS**

The *Internal Field Separator* that is used for word splitting after expansion and to split lines into words with the **read** builtin command. The default value is "<space><tab><newline>".

**IGNOREEOF**

Controls the action of an interactive shell on receipt of an **EOF** character as the sole input. If set, the value is the number of consecutive **EOF** characters which must be typed as the first characters on an input line before **bash** exits. If the variable exists but does not have a numeric value, or has no value, the default value is 10. If it does not exist, **EOF** signifies the end of input to the shell.

**INPUTRC**

The filename for the **readline** startup file, overriding the default of *~/.inputrc* (see **README** below).

**LANG** Used to determine the locale category for any category not specifically selected with a variable starting with **LC\_**.

**LC\_ALL**

This variable overrides the value of **LANG** and any other **LC\_** variable specifying a locale category.

**LC\_COLLATE**

This variable determines the collation order used when sorting the results of pathname expansion, and determines the behavior of range expressions, equivalence classes, and collating sequences within pathname expansion and pattern matching.

**LC\_CTYPE**

This variable determines the interpretation of characters and the behavior of character classes within pathname expansion and pattern matching.

**LC\_MESSAGES**

This variable determines the locale used to translate double-quoted strings preceded by a \$.

**LC\_NUMERIC**

This variable determines the locale category used for number formatting.

**LINES** Used by the **select** builtin command to determine the column length for printing selection lists. Automatically set upon receipt of a SIGWINCH.

**MAIL** If this parameter is set to a file name and the **MAILPATH** variable is not set, **bash** informs the user of the arrival of mail in the specified file.

**MAILCHECK**

Specifies how often (in seconds) **bash** checks for mail. The default is 60 seconds. When it is time to check for mail, the shell does so before displaying the primary prompt. If this variable is unset, or set to a value that is not a number greater than or equal to zero, the shell disables mail checking.

**MAILPATH**

A colon-separated list of file names to be checked for mail. The message to be printed when mail arrives in a particular file may be specified by separating the file name from the message with a '?'. When used in the text of the message, **\$\_** expands to the name of the current mailfile. Example:

```
MAILPATH=/var/mail/bfox?"You have mail":~/shell-mail?"$_ has mail!"
```

**Bash** supplies a default value for this variable, but the location of the user mail files that it uses is system dependent (e.g., /var/mail/\$USER).

#### OPTERR

If set to the value 1, **bash** displays error messages generated by the **getopts** builtin command (see **SHELL BUILTIN COMMANDS** below). **OPTERR** is initialized to 1 each time the shell is invoked or a shell script is executed.

#### PATH

The search path for commands. It is a colon-separated list of directories in which the shell looks for commands (see **COMMAND EXECUTION** below). A zero-length (null) directory name in the value of **PATH** indicates the current directory. A null directory name may appear as two adjacent colons, or as an initial or trailing colon. The default path is system-dependent, and is set by the administrator who installs **bash**. A common value is /usr/gnu/bin:/usr/local/bin:/usr/ucb:/bin:/usr/bin.

#### POSIXLY\_CORRECT

If this variable is in the environment when **bash** starts, the shell enters *posix mode* before reading the startup files, as if the **--posix** invocation option had been supplied. If it is set while the shell is running, **bash** enables *posix mode*, as if the command **set -o posix** had been executed.

#### PROMPT\_COMMAND

If set, the value is executed as a command prior to issuing each primary prompt.

#### PS1

The value of this parameter is expanded (see **PROMPTING** below) and used as the primary prompt string. The default value is "**\s-\v\\$**".

#### PS2

The value of this parameter is expanded as with **PS1** and used as the secondary prompt string. The default is ">".

#### PS3

The value of this parameter is used as the prompt for the **select** command (see **SHELL GRAMMAR** above).

#### PS4

The value of this parameter is expanded as with **PS1** and the value is printed before each command **bash** displays during an execution trace. The first character of **PS4** is replicated multiple times, as necessary, to indicate multiple levels of indirection. The default is "+".

#### SHELL

The full pathname to the shell is kept in this environment variable. If it is not set when the shell starts, **bash** assigns to it the full pathname of the current user's login shell.

#### TIMEFORMAT

The value of this parameter is used as a format string specifying how the timing information for pipelines prefixed with the **time** reserved word should be displayed. The % character introduces an escape sequence that is expanded to a time value or other information. The escape sequences and their meanings are as follows; the braces denote optional portions.

|          |                                                 |
|----------|-------------------------------------------------|
| % %      | A literal %.                                    |
| %[p][l]R | The elapsed time in seconds.                    |
| %[p][l]U | The number of CPU seconds spent in user mode.   |
| %[p][l]S | The number of CPU seconds spent in system mode. |
| %P       | The CPU percentage, computed as (%U + %S) / %R. |

The optional *p* is a digit specifying the *precision*, the number of fractional digits after a decimal point. A value of 0 causes no decimal point or fraction to be output. At most three places after the decimal point may be specified; values of *p* greater than 3 are changed to 3. If *p* is not specified, the value 3 is used.

The optional *l* specifies a longer format, including minutes, of the form *MMmSS.FFs*. The value of *p* determines whether or not the fraction is included.

If this variable is not set, **bash** acts as if it had the value **\$'\nreal\t%3lR\nuser\t%3lU\nsys\t%3lS'**. If the value is null, no timing information is displayed. A trailing newline is added when the format string is displayed.

#### TMOUT

If set to a value greater than zero, **TMOUT** is treated as the default timeout for the **read** builtin. The **select** command terminates if input does not arrive after **TMOUT** seconds when input is

coming from a terminal. In an interactive shell, the value is interpreted as the number of seconds to wait for input after issuing the primary prompt. **Bash** terminates after waiting for that number of seconds if input does not arrive.

### TMPDIR

If set, **Bash** uses its value as the name of a directory in which **Bash** creates temporary files for the shell's use.

### auto\_resume

This variable controls how the shell interacts with the user and job control. If this variable is set, single word simple commands without redirections are treated as candidates for resumption of an existing stopped job. There is no ambiguity allowed; if there is more than one job beginning with the string typed, the job most recently accessed is selected. The *name* of a stopped job, in this context, is the command line used to start it. If set to the value *exact*, the string supplied must match the name of a stopped job exactly; if set to *substring*, the string supplied needs to match a substring of the name of a stopped job. The *substring* value provides functionality analogous to the *%?* job identifier (see **JOB CONTROL** below). If set to any other value, the supplied string must be a prefix of a stopped job's name; this provides functionality analogous to the *%string* job identifier.

### histchars

The two or three characters which control history expansion and tokenization (see **HISTORY EXPANSION** below). The first character is the *history expansion* character, the character which signals the start of a history expansion, normally '!'. The second character is the *quick substitution* character, which is used as shorthand for re-running the previous command entered, substituting one string for another in the command. The default is '^'. The optional third character is the character which indicates that the remainder of the line is a comment when found as the first character of a word, normally '#'. The history comment character causes history substitution to be skipped for the remaining words on the line. It does not necessarily cause the shell parser to treat the rest of the line as a comment.

### Arrays

**Bash** provides one-dimensional array variables. Any variable may be used as an array; the **declare** builtin will explicitly declare an array. There is no maximum limit on the size of an array, nor any requirement that members be indexed or assigned contiguously. Arrays are indexed using integers and are zero-based.

An array is created automatically if any variable is assigned to using the syntax *name[subscript]=value*. The *subscript* is treated as an arithmetic expression that must evaluate to a number greater than or equal to zero. To explicitly declare an array, use **declare -a name** (see **SHELL BUILTIN COMMANDS** below). **declare -a name[subscript]** is also accepted; the *subscript* is ignored. Attributes may be specified for an array variable using the **declare** and **readonly** builtins. Each attribute applies to all members of an array.

Arrays are assigned to using compound assignments of the form *name=(value1 ... valuen)*, where each *value* is of the form *[subscript]=string*. Only *string* is required. If the optional brackets and subscript are supplied, that index is assigned to; otherwise the index of the element assigned is the last index assigned to by the statement plus one. Indexing starts at zero. This syntax is also accepted by the **declare** builtin. Individual array elements may be assigned to using the *name[subscript]=value* syntax introduced above.

Any element of an array may be referenced using *\${name[subscript]}*. The braces are required to avoid conflicts with pathname expansion. If *subscript* is @ or \*, the word expands to all members of *name*. These subscripts differ only when the word appears within double quotes. If the word is double-quoted, *\${name[\*]}* expands to a single word with the value of each array member separated by the first character of the **IFS** special variable, and *\${name[@]}* expands each element of *name* to a separate word. When there are no array members, *\${name[@]}* expands to nothing. If the double-quoted expansion occurs within a word, the expansion of the first parameter is joined with the beginning part of the original word, and the expansion of the last parameter is joined with the last part of the original word. This is analogous to the expansion of the special parameters \* and @ (see **Special Parameters** above). *\${#name[subscript]}* expands to the length of *\${name[subscript]}*. If *subscript* is \* or @, the expansion is the number of elements in the array. Referencing an array variable without a subscript is equivalent to referencing element

zero.

The **unset** builtin is used to destroy arrays. **unset** *name*[*subscript*] destroys the array element at index *subscript*. Care must be taken to avoid unwanted side effects caused by filename generation. **unset** *name*, where *name* is an array, or **unset** *name*[*subscript*], where *subscript* is \* or @, removes the entire array.

The **declare**, **local**, and **readonly** builtins each accept a **-a** option to specify an array. The **read** builtin accepts a **-a** option to assign a list of words read from the standard input to an array. The **set** and **declare** builtins display array values in a way that allows them to be reused as assignments.

## EXPANSION

Expansion is performed on the command line after it has been split into words. There are seven kinds of expansion performed: *brace expansion*, *tilde expansion*, *parameter and variable expansion*, *command substitution*, *arithmetic expansion*, *word splitting*, and *pathname expansion*.

The order of expansions is: brace expansion, tilde expansion, parameter, variable and arithmetic expansion and command substitution (done in a left-to-right fashion), word splitting, and pathname expansion.

On systems that can support it, there is an additional expansion available: *process substitution*.

Only brace expansion, word splitting, and pathname expansion can change the number of words of the expansion; other expansions expand a single word to a single word. The only exceptions to this are the expansions of "\$@" and "\${name[@]}" as explained above (see **PARAMETERS**).

### Brace Expansion

*Brace expansion* is a mechanism by which arbitrary strings may be generated. This mechanism is similar to *pathname expansion*, but the filenames generated need not exist. Patterns to be brace expanded take the form of an optional *preamble*, followed by either a series of comma-separated strings or a sequence expression between a pair of braces, followed by an optional *postscript*. The preamble is prefixed to each string contained within the braces, and the postscript is then appended to each resulting string, expanding left to right.

Brace expansions may be nested. The results of each expanded string are not sorted; left to right order is preserved. For example, `a{d,c,b}e` expands into 'ade ace abe'.

A sequence expression takes the form `{x..y}`, where *x* and *y* are either integers or single characters. When integers are supplied, the expression expands to each number between *x* and *y*, inclusive. When characters are supplied, the expression expands to each character lexicographically between *x* and *y*, inclusive. Note that both *x* and *y* must be of the same type.

Brace expansion is performed before any other expansions, and any characters special to other expansions are preserved in the result. It is strictly textual. **Bash** does not apply any syntactic interpretation to the context of the expansion or the text between the braces.

A correctly-formed brace expansion must contain unquoted opening and closing braces, and at least one unquoted comma or a valid sequence expression. Any incorrectly formed brace expansion is left unchanged. A { or , may be quoted with a backslash to prevent its being considered part of a brace expression. To avoid conflicts with parameter expansion, the string \${ is not considered eligible for brace expansion.

This construct is typically used as shorthand when the common prefix of the strings to be generated is longer than in the above example:

```
mkdir /usr/local/src/bash/{old,new,dist,bugs}
```

or

```
chown root /usr/{ucb/{ex,edit},lib/{ex?.?*,how_ex}}
```

Brace expansion introduces a slight incompatibility with historical versions of **sh**. **sh** does not treat opening or closing braces specially when they appear as part of a word, and preserves them in the output. **Bash** removes braces from words as a consequence of brace expansion. For example, a word entered to **sh** as `file{1,2}` appears identically in the output. The same word is output as `file1 file2` after expansion by **bash**. If strict compatibility with **sh** is desired, start **bash** with the **+B** option or disable brace expansion with the **+B** option to the **set** command (see **SHELL BUILTIN COMMANDS** below).

### Tilde Expansion

If a word begins with an unquoted tilde character ('~'), all of the characters preceding the first unquoted slash (or all characters, if there is no unquoted slash) are considered a *tilde-prefix*. If none of the characters in the tilde-prefix are quoted, the characters in the tilde-prefix following the tilde are treated as a possible *login name*. If this login name is the null string, the tilde is replaced with the value of the shell parameter **HOME**. If **HOME** is unset, the home directory of the user executing the shell is substituted instead. Otherwise, the tilde-prefix is replaced with the home directory associated with the specified login name.

If the tilde-prefix is a '~+', the value of the shell variable **PWD** replaces the tilde-prefix. If the tilde-prefix is a '~-', the value of the shell variable **OLDPWD**, if it is set, is substituted. If the characters following the tilde in the tilde-prefix consist of a number *N*, optionally prefixed by a '+' or a '-', the tilde-prefix is replaced with the corresponding element from the directory stack, as it would be displayed by the **dirs** builtin invoked with the tilde-prefix as an argument. If the characters following the tilde in the tilde-prefix consist of a number without a leading '+' or '-', '+' is assumed.

If the login name is invalid, or the tilde expansion fails, the word is unchanged.

Each variable assignment is checked for unquoted tilde-prefixes immediately following a ':' or the first '='. In these cases, tilde expansion is also performed. Consequently, one may use file names with tildes in assignments to **PATH**, **MAILPATH**, and **CDPATH**, and the shell assigns the expanded value.

### Parameter Expansion

The '\$' character introduces parameter expansion, command substitution, or arithmetic expansion. The parameter name or symbol to be expanded may be enclosed in braces, which are optional but serve to protect the variable to be expanded from characters immediately following it which could be interpreted as part of the name.

When braces are used, the matching ending brace is the first '}' not escaped by a backslash or within a quoted string, and not within an embedded arithmetic expansion, command substitution, or parameter expansion.

`${parameter}`

The value of *parameter* is substituted. The braces are required when *parameter* is a positional parameter with more than one digit, or when *parameter* is followed by a character which is not to be interpreted as part of its name.

If the first character of *parameter* is an exclamation point, a level of variable indirection is introduced. **Bash** uses the value of the variable formed from the rest of *parameter* as the name of the variable; this variable is then expanded and that value is used in the rest of the substitution, rather than the value of *parameter* itself. This is known as *indirect expansion*. The exceptions to this are the expansions of `${!prefix*}` and `${!name[@]}` described below. The exclamation point must immediately follow the left brace in order to introduce indirection.

In each of the cases below, *word* is subject to tilde expansion, parameter expansion, command substitution, and arithmetic expansion. When not performing substring expansion, **bash** tests for a parameter that is unset or null; omitting the colon results in a test only for a parameter that is unset.

`${parameter:-word}`

**Use Default Values.** If *parameter* is unset or null, the expansion of *word* is substituted. Otherwise, the value of *parameter* is substituted.

`${parameter:=word}`

**Assign Default Values.** If *parameter* is unset or null, the expansion of *word* is assigned to *parameter*. The value of *parameter* is then substituted. Positional parameters and special parameters may not be assigned to in this way.

`${parameter:?word}`

**Display Error if Null or Unset.** If *parameter* is null or unset, the expansion of *word* (or a message to that effect if *word* is not present) is written to the standard error and the shell, if it is not interactive, exits. Otherwise, the value of *parameter* is substituted.

`${parameter:+word}`

**Use Alternate Value.** If *parameter* is null or unset, nothing is substituted, otherwise the expansion of *word* is substituted.

`${parameter:offset}`

`${parameter:offset:length}`

**Substring Expansion.** Expands to up to *length* characters of *parameter* starting at the character specified by *offset*. If *length* is omitted, expands to the substring of *parameter* starting at the character specified by *offset*. *length* and *offset* are arithmetic expressions (see **ARITHMETIC EVALUATION** below). *length* must evaluate to a number greater than or equal to zero. If *offset* evaluates to a number less than zero, the value is used as an offset from the end of the value of *parameter*. If *parameter* is `@`, the result is *length* positional parameters beginning at *offset*. If *parameter* is an array name indexed by `@` or `*`, the result is the *length* members of the array beginning with `${parameter[offset]}`. A negative *offset* is taken relative to one greater than the maximum index of the specified array. Note that a negative offset must be separated from the colon by at least one space to avoid being confused with the `:-` expansion. Substring indexing is zero-based unless the positional parameters are used, in which case the indexing starts at 1.

`${!prefix*}`

`${!prefix@}`

Expands to the names of variables whose names begin with *prefix*, separated by the first character of the **IFS** special variable.

`${!name[@]}`

`${!name[*]}`

If *name* is an array variable, expands to the list of array indices (keys) assigned in *name*. If *name* is not an array, expands to 0 if *name* is set and null otherwise. When `@` is used and the expansion appears within double quotes, each key expands to a separate word.

`${#parameter}`

The length in characters of the value of *parameter* is substituted. If *parameter* is `*` or `@`, the value substituted is the number of positional parameters. If *parameter* is an array name subscripted by `*` or `@`, the value substituted is the number of elements in the array.

`${parameter#word}`

`${parameter##word}`

The *word* is expanded to produce a pattern just as in pathname expansion. If the pattern matches the beginning of the value of *parameter*, then the result of the expansion is the expanded value of *parameter* with the shortest matching pattern (the `"#"` case) or the longest matching pattern (the `"##"` case) deleted. If *parameter* is `@` or `*`, the pattern removal operation is applied to each positional parameter in turn, and the expansion is the resultant list. If *parameter* is an array variable subscripted with `@` or `*`, the pattern removal operation is applied to each member of the array in turn, and the expansion is the resultant list.

`${parameter%word}`

`${parameter%%word}`

The *word* is expanded to produce a pattern just as in pathname expansion. If the pattern matches a trailing portion of the expanded value of *parameter*, then the result of the expansion is the expanded value of *parameter* with the shortest matching pattern (the `"%"` case) or the longest matching pattern (the `"%%"` case) deleted. If *parameter* is `@` or `*`, the pattern removal operation is applied to each positional parameter in turn, and the expansion is the resultant list. If *parameter* is an array variable subscripted with `@` or `*`, the pattern removal operation is applied to each member of the array in turn, and the expansion is the resultant list.

`${parameter/pattern/string}`

The *pattern* is expanded to produce a pattern just as in pathname expansion. *Parameter* is expanded and the longest match of *pattern* against its value is replaced with *string*. If *pattern* begins with `/`, all matches of *pattern* are replaced with *string*. Normally only the first match is replaced. If *pattern* begins with `#`, it must match at the beginning of the expanded value of

*parameter*. If *pattern* begins with %, it must match at the end of the expanded value of *parameter*. If *string* is null, matches of *pattern* are deleted and the *l* following *pattern* may be omitted. If *parameter* is @ or \*, the substitution operation is applied to each positional parameter in turn, and the expansion is the resultant list. If *parameter* is an array variable subscripted with @ or \*, the substitution operation is applied to each member of the array in turn, and the expansion is the resultant list.

### Command Substitution

*Command substitution* allows the output of a command to replace the command name. There are two forms:

`$(command)`

or

``command``

**Bash** performs the expansion by executing *command* and replacing the command substitution with the standard output of the command, with any trailing newlines deleted. Embedded newlines are not deleted, but they may be removed during word splitting. The command substitution `$(cat file)` can be replaced by the equivalent but faster `$(<file)`.

When the old-style backquote form of substitution is used, backslash retains its literal meaning except when followed by \$, `, or \. The first backquote not preceded by a backslash terminates the command substitution. When using the `$(command)` form, all characters between the parentheses make up the command; none are treated specially.

Command substitutions may be nested. To nest when using the backquoted form, escape the inner backquotes with backslashes.

If the substitution appears within double quotes, word splitting and pathname expansion are not performed on the results.

### Arithmetic Expansion

Arithmetic expansion allows the evaluation of an arithmetic expression and the substitution of the result. The format for arithmetic expansion is:

`$((expression))`

The *expression* is treated as if it were within double quotes, but a double quote inside the parentheses is not treated specially. All tokens in the expression undergo parameter expansion, string expansion, command substitution, and quote removal. Arithmetic expansions may be nested.

The evaluation is performed according to the rules listed below under **ARITHMETIC EVALUATION**. If *expression* is invalid, **bash** prints a message indicating failure and no substitution occurs.

### Process Substitution

*Process substitution* is supported on systems that support named pipes (*FIFOs*) or the `/dev/fd` method of naming open files. It takes the form of `<(list)` or `>(list)`. The process *list* is run with its input or output connected to a *FIFO* or some file in `/dev/fd`. The name of this file is passed as an argument to the current command as the result of the expansion. If the `>(list)` form is used, writing to the file will provide input for *list*. If the `<(list)` form is used, the file passed as an argument should be read to obtain the output of *list*.

When available, process substitution is performed simultaneously with parameter and variable expansion, command substitution, and arithmetic expansion.

### Word Splitting

The shell scans the results of parameter expansion, command substitution, and arithmetic expansion that did not occur within double quotes for *word splitting*.

The shell treats each character of **IFS** as a delimiter, and splits the results of the other expansions into words on these characters. If **IFS** is unset, or its value is exactly `<space><tab><newline>`, the default, then any sequence of **IFS** characters serves to delimit words. If **IFS** has a value other than the default, then sequences of the whitespace characters **space** and **tab** are ignored at the beginning and end of the word, as long as the whitespace character is in the value of **IFS** (an **IFS** whitespace character). Any character in **IFS**



that is not **IFS** whitespace, along with any adjacent **IFS** whitespace characters, delimits a field. A sequence of **IFS** whitespace characters is also treated as a delimiter. If the value of **IFS** is null, no word splitting occurs.

Explicit null arguments ("" or '') are retained. Unquoted implicit null arguments, resulting from the expansion of parameters that have no values, are removed. If a parameter with no value is expanded within double quotes, a null argument results and is retained.

Note that if no expansion occurs, no splitting is performed.

### Pathname Expansion

After word splitting, unless the **-f** option has been set, **bash** scans each word for the characters **\***, **?**, and **[**. If one of these characters appears, then the word is regarded as a *pattern*, and replaced with an alphabetically sorted list of file names matching the pattern. If no matching file names are found, and the shell option **nullglob** is disabled, the word is left unchanged. If the **nullglob** option is set, and no matches are found, the word is removed. If the **failglob** shell option is set, and no matches are found, an error message is printed and the command is not executed. If the shell option **nocaseglob** is enabled, the match is performed without regard to the case of alphabetic characters. When a pattern is used for pathname expansion, the character **"."** at the start of a name or immediately following a slash must be matched explicitly, unless the shell option **dotglob** is set. When matching a pathname, the slash character must always be matched explicitly. In other cases, the **"."** character is not treated specially. See the description of **shopt** below under **SHELL BUILTIN COMMANDS** for a description of the **nocaseglob**, **nullglob**, **failglob**, and **dotglob** shell options.

The **GLOBIGNORE** shell variable may be used to restrict the set of file names matching a *pattern*. If **GLOBIGNORE** is set, each matching file name that also matches one of the patterns in **GLOBIGNORE** is removed from the list of matches. The file names **"."** and **".."** are always ignored when **GLOBIGNORE** is set and not null. However, setting **GLOBIGNORE** to a non-null value has the effect of enabling the **dotglob** shell option, so all other file names beginning with a **"."** will match. To get the old behavior of ignoring file names beginning with a **"."**, make **"\*."** one of the patterns in **GLOBIGNORE**. The **dotglob** option is disabled when **GLOBIGNORE** is unset.

### Pattern Matching

Any character that appears in a pattern, other than the special pattern characters described below, matches itself. The NUL character may not occur in a pattern. A backslash escapes the following character; the escaping backslash is discarded when matching. The special pattern characters must be quoted if they are to be matched literally.

The special pattern characters have the following meanings:

- \*** Matches any string, including the null string.
- ?** Matches any single character.
- [...]** Matches any one of the enclosed characters. A pair of characters separated by a hyphen denotes a *range expression*; any character that sorts between those two characters, inclusive, using the current locale's collating sequence and character set, is matched. If the first character following the **[** is a **!** or a **^** then any character not enclosed is matched. The sorting order of characters in range expressions is determined by the current locale and the value of the **LC\_COLLATE** shell variable, if set. A **-** may be matched by including it as the first or last character in the set. A **]** may be matched by including it as the first character in the set.

Within **[** and **]**, *character classes* can be specified using the syntax **[*class*]**, where *class* is one of the following classes defined in the POSIX standard:

**alnum alpha ascii blank cntrl digit graph lower print punct space upper word xdigit**

A character class matches any character belonging to that class. The **word** character class matches letters, digits, and the character **\_**.

Within **[** and **]**, an *equivalence class* can be specified using the syntax **[*=c*]**, which matches all characters with the same collation weight (as defined by the current locale) as the character *c*.

Within [ and ], the syntax [*symbol*.] matches the collating symbol *symbol*.

If the **extglob** shell option is enabled using the **shopt** builtin, several extended pattern matching operators are recognized. In the following description, a *pattern-list* is a list of one or more patterns separated by a |. Composite patterns may be formed using one or more of the following sub-patterns:

```
?(pattern-list)
 Matches zero or one occurrence of the given patterns
*(pattern-list)
 Matches zero or more occurrences of the given patterns
+(pattern-list)
 Matches one or more occurrences of the given patterns
@(pattern-list)
 Matches one of the given patterns
!(pattern-list)
 Matches anything except one of the given patterns
```

### Quote Removal

After the preceding expansions, all unquoted occurrences of the characters \, ', and " that did not result from one of the above expansions are removed.

### REDIRECTION

Before a command is executed, its input and output may be *redirected* using a special notation interpreted by the shell. Redirection may also be used to open and close files for the current shell execution environment. The following redirection operators may precede or appear anywhere within a *simple command* or may follow a *command*. Redirections are processed in the order they appear, from left to right.

In the following descriptions, if the file descriptor number is omitted, and the first character of the redirection operator is <, the redirection refers to the standard input (file descriptor 0). If the first character of the redirection operator is >, the redirection refers to the standard output (file descriptor 1).

The word following the redirection operator in the following descriptions, unless otherwise noted, is subjected to brace expansion, tilde expansion, parameter expansion, command substitution, arithmetic expansion, quote removal, pathname expansion, and word splitting. If it expands to more than one word, **bash** reports an error.

Note that the order of redirections is significant. For example, the command

```
ls > dirlist 2>&1
```

directs both standard output and standard error to the file *dirlist*, while the command

```
ls 2>&1 > dirlist
```

directs only the standard output to file *dirlist*, because the standard error was duplicated as standard output before the standard output was redirected to *dirlist*.

**Bash** handles several filenames specially when they are used in redirections, as described in the following table:

```
/dev/fd/fd
 If fd is a valid integer, file descriptor fd is duplicated.
/dev/stdin
 File descriptor 0 is duplicated.
/dev/stdout
 File descriptor 1 is duplicated.
/dev/stderr
 File descriptor 2 is duplicated.
/dev/tcp/host/port
 If host is a valid hostname or Internet address, and port is an integer port number or service name, bash attempts to open a TCP connection to the corresponding socket.
```

**/dev/udp/host/port**

If *host* is a valid hostname or Internet address, and *port* is an integer port number or service name, **bash** attempts to open a UDP connection to the corresponding socket.

A failure to open or create a file causes the redirection to fail.

Redirections using file descriptors greater than 9 should be used with care, as they may conflict with file descriptors the shell uses internally.

### Redirecting Input

Redirection of input causes the file whose name results from the expansion of *word* to be opened for reading on file descriptor *n*, or the standard input (file descriptor 0) if *n* is not specified.

The general format for redirecting input is:

**[n]<word**

### Redirecting Output

Redirection of output causes the file whose name results from the expansion of *word* to be opened for writing on file descriptor *n*, or the standard output (file descriptor 1) if *n* is not specified. If the file does not exist it is created; if it does exist it is truncated to zero size.

The general format for redirecting output is:

**[n]>word**

If the redirection operator is **>**, and the **noclobber** option to the **set** builtin has been enabled, the redirection will fail if the file whose name results from the expansion of *word* exists and is a regular file. If the redirection operator is **>|**, or the redirection operator is **>** and the **noclobber** option to the **set** builtin command is not enabled, the redirection is attempted even if the file named by *word* exists.

### Appending Redirected Output

Redirection of output in this fashion causes the file whose name results from the expansion of *word* to be opened for appending on file descriptor *n*, or the standard output (file descriptor 1) if *n* is not specified. If the file does not exist it is created.

The general format for appending output is:

**[n]>>word**

### Redirecting Standard Output and Standard Error

**Bash** allows both the standard output (file descriptor 1) and the standard error output (file descriptor 2) to be redirected to the file whose name is the expansion of *word* with this construct.

There are two formats for redirecting standard output and standard error:

**&>word**

and

**>&word**

Of the two forms, the first is preferred. This is semantically equivalent to

**>word 2>&1**

### Here Documents

This type of redirection instructs the shell to read input from the current source until a line containing only *word* (with no trailing blanks) is seen. All of the lines read up to that point are then used as the standard input for a command.

The format of here-documents is:

**<<[-]word**  
*here-document*  
*delimiter*

No parameter expansion, command substitution, arithmetic expansion, or pathname expansion is performed on *word*. If any characters in *word* are quoted, the *delimiter* is the result of quote removal on *word*, and

the lines in the here-document are not expanded. If *word* is unquoted, all lines of the here-document are subjected to parameter expansion, command substitution, and arithmetic expansion. In the latter case, the character sequence `\<newline>` is ignored, and `\` must be used to quote the characters `\`, `$`, and `'`.

If the redirection operator is `<<-`, then all leading tab characters are stripped from input lines and the line containing *delimiter*. This allows here-documents within shell scripts to be indented in a natural fashion.

### Here Strings

A variant of here documents, the format is:

```
<<<word
```

The *word* is expanded and supplied to the command on its standard input.

### Duplicating File Descriptors

The redirection operator

```
[n]<&word
```

is used to duplicate input file descriptors. If *word* expands to one or more digits, the file descriptor denoted by *n* is made to be a copy of that file descriptor. If the digits in *word* do not specify a file descriptor open for input, a redirection error occurs. If *word* evaluates to `-`, file descriptor *n* is closed. If *n* is not specified, the standard input (file descriptor 0) is used.

The operator

```
[n]>&word
```

is used similarly to duplicate output file descriptors. If *n* is not specified, the standard output (file descriptor 1) is used. If the digits in *word* do not specify a file descriptor open for output, a redirection error occurs. As a special case, if *n* is omitted, and *word* does not expand to one or more digits, the standard output and standard error are redirected as described previously.

### Moving File Descriptors

The redirection operator

```
[n]<&digit-
```

moves the file descriptor *digit* to file descriptor *n*, or the standard input (file descriptor 0) if *n* is not specified. *digit* is closed after being duplicated to *n*.

Similarly, the redirection operator

```
[n]>&digit-
```

moves the file descriptor *digit* to file descriptor *n*, or the standard output (file descriptor 1) if *n* is not specified.

### Opening File Descriptors for Reading and Writing

The redirection operator

```
[n]<>word
```

causes the file whose name is the expansion of *word* to be opened for both reading and writing on file descriptor *n*, or on file descriptor 0 if *n* is not specified. If the file does not exist, it is created.

## ALIASES

*Aliases* allow a string to be substituted for a word when it is used as the first word of a simple command. The shell maintains a list of aliases that may be set and unset with the **alias** and **unalias** builtin commands (see **SHELL BUILTIN COMMANDS** below). The first word of each simple command, if unquoted, is checked to see if it has an alias. If so, that word is replaced by the text of the alias. The characters `/`, `$`, `'`, and `=` and any of the shell *metacharacters* or quoting characters listed above may not appear in an alias name. The replacement text may contain any valid shell input, including shell metacharacters. The first word of the replacement text is tested for aliases, but a word that is identical to an alias being expanded is not expanded a second time. This means that one may alias **ls** to **ls -F**, for instance, and **bash** does not try to recursively expand the replacement text. If the last character of the alias value is a *blank*, then the next

command word following the alias is also checked for alias expansion.

Aliases are created and listed with the **alias** command, and removed with the **unalias** command.

There is no mechanism for using arguments in the replacement text. If arguments are needed, a shell function should be used (see **FUNCTIONS** below).

Aliases are not expanded when the shell is not interactive, unless the **expand\_aliases** shell option is set using **shopt** (see the description of **shopt** under **SHELL BUILTIN COMMANDS** below).

The rules concerning the definition and use of aliases are somewhat confusing. **Bash** always reads at least one complete line of input before executing any of the commands on that line. Aliases are expanded when a command is read, not when it is executed. Therefore, an alias definition appearing on the same line as another command does not take effect until the next line of input is read. The commands following the alias definition on that line are not affected by the new alias. This behavior is also an issue when functions are executed. Aliases are expanded when a function definition is read, not when the function is executed, because a function definition is itself a compound command. As a consequence, aliases defined in a function are not available until after that function is executed. To be safe, always put alias definitions on a separate line, and do not use **alias** in compound commands.

For almost every purpose, aliases are superseded by shell functions.

## FUNCTIONS

A shell function, defined as described above under **SHELL GRAMMAR**, stores a series of commands for later execution. When the name of a shell function is used as a simple command name, the list of commands associated with that function name is executed. Functions are executed in the context of the current shell; no new process is created to interpret them (contrast this with the execution of a shell script). When a function is executed, the arguments to the function become the positional parameters during its execution. The special parameter **#** is updated to reflect the change. Special parameter **0** is unchanged. The first element of the **FUNCNAME** variable is set to the name of the function while the function is executing. All other aspects of the shell execution environment are identical between a function and its caller with the exception that the **DEBUG** and **RETURN** traps (see the description of the **trap** builtin under **SHELL BUILTIN COMMANDS** below) are not inherited unless the function has been given the **trace** attribute (see the description of the **declare** builtin below) or the **-o functrace** shell option has been enabled with the **set** builtin (in which case all functions inherit the **DEBUG** and **RETURN** traps).

Variables local to the function may be declared with the **local** builtin command. Ordinarily, variables and their values are shared between the function and its caller.

If the builtin command **return** is executed in a function, the function completes and execution resumes with the next command after the function call. Any command associated with the **RETURN** trap is executed before execution resumes. When a function completes, the values of the positional parameters and the special parameter **#** are restored to the values they had prior to the function's execution.

Function names and definitions may be listed with the **-f** option to the **declare** or **typeset** builtin commands. The **-F** option to **declare** or **typeset** will list the function names only (and optionally the source file and line number, if the **extdebug** shell option is enabled). Functions may be exported so that subshells automatically have them defined with the **-f** option to the **export** builtin. A function definition may be deleted using the **-f** option to the **unset** builtin. Note that shell functions and variables with the same name may result in multiple identically-named entries in the environment passed to the shell's children. Care should be taken in cases where this may cause a problem.

Functions may be recursive. No limit is imposed on the number of recursive calls.

## ARITHMETIC EVALUATION

The shell allows arithmetic expressions to be evaluated, under certain circumstances (see the **let** and **declare** builtin commands and **Arithmetic Expansion**). Evaluation is done in fixed-width integers with no check for overflow, though division by 0 is trapped and flagged as an error. The operators and their precedence, associativity, and values are the same as in the C language. The following list of operators is grouped into levels of equal-precedence operators. The levels are listed in order of decreasing precedence.

```

id++ id--
 variable post-increment and post-decrement
++id --id
 variable pre-increment and pre-decrement
- +
 unary minus and plus
! ~
 logical and bitwise negation
**
 exponentiation
* / %
 multiplication, division, remainder
+ -
 addition, subtraction
<< >>
 left and right bitwise shifts
<= >= < >
 comparison
== !=
 equality and inequality
&
 bitwise AND
^
 bitwise exclusive OR
|
 bitwise OR
&&
 logical AND
||
 logical OR
expr?expr:expr
 conditional operator
= *= /= %= += -= <<= >>= &= ^= |=
 assignment
expr1 , expr2
 comma

```

Shell variables are allowed as operands; parameter expansion is performed before the expression is evaluated. Within an expression, shell variables may also be referenced by name without using the parameter expansion syntax. A shell variable that is null or unset evaluates to 0 when referenced by name without using the parameter expansion syntax. The value of a variable is evaluated as an arithmetic expression when it is referenced, or when a variable which has been given the *integer* attribute using **declare -i** is assigned a value. A null value evaluates to 0. A shell variable need not have its integer attribute turned on to be used in an expression.

Constants with a leading 0 are interpreted as octal numbers. A leading 0x or 0X denotes hexadecimal. Otherwise, numbers take the form [*base#*]n, where *base* is a decimal number between 2 and 64 representing the arithmetic base, and *n* is a number in that base. If *base#* is omitted, then base 10 is used. The digits greater than 9 are represented by the lowercase letters, the uppercase letters, @, and \_, in that order. If *base* is less than or equal to 36, lowercase and uppercase letters may be used interchangeably to represent numbers between 10 and 35.

Operators are evaluated in order of precedence. Sub-expressions in parentheses are evaluated first and may override the precedence rules above.

## CONDITIONAL EXPRESSIONS

Conditional expressions are used by the **[[** compound command and the **test** and **[** builtin commands to test file attributes and perform string and arithmetic comparisons. Expressions are formed from the following unary or binary primaries. If any *file* argument to one of the primaries is of the form */dev/fd/n*, then file descriptor *n* is checked. If the *file* argument to one of the primaries is one of */dev/stdin*, */dev/stdout*, or */dev/stderr*, file descriptor 0, 1, or 2, respectively, is checked.

Unless otherwise specified, primaries that operate on files follow symbolic links and operate on the target of the link, rather than the link itself.

```

-a file True if file exists.
-b file True if file exists and is a block special file.
-c file True if file exists and is a character special file.

```

**-d file** True if *file* exists and is a directory.  
**-e file** True if *file* exists.  
**-f file** True if *file* exists and is a regular file.  
**-g file** True if *file* exists and is set-group-id.  
**-h file** True if *file* exists and is a symbolic link.  
**-k file** True if *file* exists and its “sticky” bit is set.  
**-p file** True if *file* exists and is a named pipe (FIFO).  
**-r file** True if *file* exists and is readable.  
**-s file** True if *file* exists and has a size greater than zero.  
**-t fd** True if file descriptor *fd* is open and refers to a terminal.  
**-u file** True if *file* exists and its set-user-id bit is set.  
**-w file** True if *file* exists and is writable.  
**-x file** True if *file* exists and is executable.  
**-O file** True if *file* exists and is owned by the effective user id.  
**-G file** True if *file* exists and is owned by the effective group id.  
**-L file** True if *file* exists and is a symbolic link.  
**-S file** True if *file* exists and is a socket.  
**-N file** True if *file* exists and has been modified since it was last read.  
**file1 -nt file2**  
 True if *file1* is newer (according to modification date) than *file2*, or if *file1* exists and *file2* does not.  
**file1 -ot file2**  
 True if *file1* is older than *file2*, or if *file2* exists and *file1* does not.  
**file1 -ef file2**  
 True if *file1* and *file2* refer to the same device and inode numbers.  
**-o optname**  
 True if shell option *optname* is enabled. See the list of options under the description of the **-o** option to the **set** builtin below.  
**-z string**  
 True if the length of *string* is zero.  
*string*  
**-n string**  
 True if the length of *string* is non-zero.  
*string1* == *string2*  
 True if the strings are equal. = may be used in place of == for strict POSIX compliance.  
*string1* != *string2*  
 True if the strings are not equal.  
*string1* < *string2*  
 True if *string1* sorts before *string2* lexicographically in the current locale.  
*string1* > *string2*  
 True if *string1* sorts after *string2* lexicographically in the current locale.  
*arg1* **OP** *arg2*  
**OP** is one of **-eq**, **-ne**, **-lt**, **-le**, **-gt**, or **-ge**. These arithmetic binary operators return true if *arg1* is equal to, not equal to, less than, less than or equal to, greater than, or greater than or equal to *arg2*, respectively. *Arg1* and *arg2* may be positive or negative integers.

## SIMPLE COMMAND EXPANSION

When a simple command is executed, the shell performs the following expansions, assignments, and redirections, from left to right.

1. The words that the parser has marked as variable assignments (those preceding the command name) and redirections are saved for later processing.
2. The words that are not variable assignments or redirections are expanded. If any words remain after expansion, the first word is taken to be the name of the command and the remaining words

are the arguments.

3. Redirections are performed as described above under **REDIRECTION**.
4. The text after the = in each variable assignment undergoes tilde expansion, parameter expansion, command substitution, arithmetic expansion, and quote removal before being assigned to the variable.

If no command name results, the variable assignments affect the current shell environment. Otherwise, the variables are added to the environment of the executed command and do not affect the current shell environment. If any of the assignments attempts to assign a value to a readonly variable, an error occurs, and the command exits with a non-zero status.

If no command name results, redirections are performed, but do not affect the current shell environment. A redirection error causes the command to exit with a non-zero status.

If there is a command name left after expansion, execution proceeds as described below. Otherwise, the command exits. If one of the expansions contained a command substitution, the exit status of the command is the exit status of the last command substitution performed. If there were no command substitutions, the command exits with a status of zero.

## COMMAND EXECUTION

After a command has been split into words, if it results in a simple command and an optional list of arguments, the following actions are taken.

If the command name contains no slashes, the shell attempts to locate it. If there exists a shell function by that name, that function is invoked as described above in **FUNCTIONS**. If the name does not match a function, the shell searches for it in the list of shell builtins. If a match is found, that builtin is invoked.

If the name is neither a shell function nor a builtin, and contains no slashes, **bash** searches each element of the **PATH** for a directory containing an executable file by that name. **Bash** uses a hash table to remember the full pathnames of executable files (see **hash** under **SHELL BUILTIN COMMANDS** below). A full search of the directories in **PATH** is performed only if the command is not found in the hash table. If the search is unsuccessful, the shell prints an error message and returns an exit status of 127.

If the search is successful, or if the command name contains one or more slashes, the shell executes the named program in a separate execution environment. Argument 0 is set to the name given, and the remaining arguments to the command are set to the arguments given, if any.

If this execution fails because the file is not in executable format, and the file is not a directory, it is assumed to be a *shell script*, a file containing shell commands. A subshell is spawned to execute it. This subshell reinitializes itself, so that the effect is as if a new shell had been invoked to handle the script, with the exception that the locations of commands remembered by the parent (see **hash** below under **SHELL BUILTIN COMMANDS**) are retained by the child.

If the program is a file beginning with **#!**, the remainder of the first line specifies an interpreter for the program. The shell executes the specified interpreter on operating systems that do not handle this executable format themselves. The arguments to the interpreter consist of a single optional argument following the interpreter name on the first line of the program, followed by the name of the program, followed by the command arguments, if any.

## COMMAND EXECUTION ENVIRONMENT

The shell has an *execution environment*, which consists of the following:

- open files inherited by the shell at invocation, as modified by redirections supplied to the **exec** builtin
- the current working directory as set by **cd**, **pushd**, or **popd**, or inherited by the shell at invocation
- the file creation mode mask as set by **umask** or inherited from the shell's parent
- current traps set by **trap**



- shell parameters that are set by variable assignment or with **set** or inherited from the shell's parent in the environment
- shell functions defined during execution or inherited from the shell's parent in the environment
- options enabled at invocation (either by default or with command-line arguments) or by **set**
- options enabled by **shopt**
- shell aliases defined with **alias**
- various process IDs, including those of background jobs, the value of **\$\$**, and the value of **\$PPID**

When a simple command other than a builtin or shell function is to be executed, it is invoked in a separate execution environment that consists of the following. Unless otherwise noted, the values are inherited from the shell.

- the shell's open files, plus any modifications and additions specified by redirections to the command
- the current working directory
- the file creation mode mask
- shell variables and functions marked for export, along with variables exported for the command, passed in the environment
- traps caught by the shell are reset to the values inherited from the shell's parent, and traps ignored by the shell are ignored

A command invoked in this separate environment cannot affect the shell's execution environment.

Command substitution, commands grouped with parentheses, and asynchronous commands are invoked in a subshell environment that is a duplicate of the shell environment, except that traps caught by the shell are reset to the values that the shell inherited from its parent at invocation. Builtin commands that are invoked as part of a pipeline are also executed in a subshell environment. Changes made to the subshell environment cannot affect the shell's execution environment.

If a command is followed by a **&** and job control is not active, the default standard input for the command is the empty file */dev/null*. Otherwise, the invoked command inherits the file descriptors of the calling shell as modified by redirections.

## ENVIRONMENT

When a program is invoked it is given an array of strings called the *environment*. This is a list of *name=value* pairs, of the form *name=value*.

The shell provides several ways to manipulate the environment. On invocation, the shell scans its own environment and creates a parameter for each name found, automatically marking it for *export* to child processes. Executed commands inherit the environment. The **export** and **declare -x** commands allow parameters and functions to be added to and deleted from the environment. If the value of a parameter in the environment is modified, the new value becomes part of the environment, replacing the old. The environment inherited by any executed command consists of the shell's initial environment, whose values may be modified in the shell, less any pairs removed by the **unset** command, plus any additions via the **export** and **declare -x** commands.

The environment for any *simple command* or function may be augmented temporarily by prefixing it with parameter assignments, as described above in **PARAMETERS**. These assignment statements affect only the environment seen by that command.

If the **-k** option is set (see the **set** builtin command below), then *all* parameter assignments are placed in the environment for a command, not just those that precede the command name.

When **bash** invokes an external command, the variable **\_** is set to the full file name of the command and passed to that command in its environment.

## EXIT STATUS

For the shell's purposes, a command which exits with a zero exit status has succeeded. An exit status of zero indicates success. A non-zero exit status indicates failure. When a command terminates on a fatal signal *N*, **bash** uses the value of  $128+N$  as the exit status.

If a command is not found, the child process created to execute it returns a status of 127. If a command is found but is not executable, the return status is 126.

If a command fails because of an error during expansion or redirection, the exit status is greater than zero.

Shell builtin commands return a status of 0 (*true*) if successful, and non-zero (*false*) if an error occurs while they execute. All builtins return an exit status of 2 to indicate incorrect usage.

**Bash** itself returns the exit status of the last command executed, unless a syntax error occurs, in which case it exits with a non-zero value. See also the **exit** builtin command below.

## SIGNALS

When **bash** is interactive, in the absence of any traps, it ignores **SIGTERM** (so that **kill 0** does not kill an interactive shell), and **SIGINT** is caught and handled (so that the **wait** builtin is interruptible). In all cases, **bash** ignores **SIGQUIT**. If job control is in effect, **bash** ignores **SIGTTIN**, **SIGTTOU**, and **SIGTSTP**.

Non-builtin commands run by **bash** have signal handlers set to the values inherited by the shell from its parent. When job control is not in effect, asynchronous commands ignore **SIGINT** and **SIGQUIT** in addition to these inherited handlers. Commands run as a result of command substitution ignore the keyboard-generated job control signals **SIGTTIN**, **SIGTTOU**, and **SIGTSTP**.

The shell exits by default upon receipt of a **SIGHUP**. Before exiting, an interactive shell resends the **SIGHUP** to all jobs, running or stopped. Stopped jobs are sent **SIGCONT** to ensure that they receive the **SIGHUP**. To prevent the shell from sending the signal to a particular job, it should be removed from the jobs table with the **disown** builtin (see **SHELL BUILTIN COMMANDS** below) or marked to not receive **SIGHUP** using **disown -h**.

If the **huponexit** shell option has been set with **shopt**, **bash** sends a **SIGHUP** to all jobs when an interactive login shell exits.

If **bash** is waiting for a command to complete and receives a signal for which a trap has been set, the trap will not be executed until the command completes. When **bash** is waiting for an asynchronous command via the **wait** builtin, the reception of a signal for which a trap has been set will cause the **wait** builtin to return immediately with an exit status greater than 128, immediately after which the trap is executed.

## JOB CONTROL

*Job control* refers to the ability to selectively stop (*suspend*) the execution of processes and continue (*resume*) their execution at a later point. A user typically employs this facility via an interactive interface supplied jointly by the system's terminal driver and **bash**.

The shell associates a *job* with each pipeline. It keeps a table of currently executing jobs, which may be listed with the **jobs** command. When **bash** starts a job asynchronously (in the *background*), it prints a line that looks like:

```
[1] 25647
```

indicating that this job is job number 1 and that the process ID of the last process in the pipeline associated with this job is 25647. All of the processes in a single pipeline are members of the same job. **Bash** uses the *job* abstraction as the basis for job control.

To facilitate the implementation of the user interface to job control, the operating system maintains the notion of a *current terminal process group ID*. Members of this process group (processes whose process group ID is equal to the current terminal process group ID) receive keyboard-generated signals such as **SIGINT**. These processes are said to be in the *foreground*. *Background* processes are those whose process group ID differs from the terminal's; such processes are immune to keyboard-generated signals. Only foreground processes are allowed to read from or write to the terminal. Background processes which attempt to read from (write to) the terminal are sent a **SIGTTIN** (**SIGTTOU**) signal by the terminal driver, which, unless caught, suspends the process.

If the operating system on which **bash** is running supports job control, **bash** contains facilities to use it. Typing the *suspend* character (typically **^Z**, Control-Z) while a process is running causes that process to be stopped and returns control to **bash**. Typing the *delayed suspend* character (typically **^Y**, Control-Y) causes the process to be stopped when it attempts to read input from the terminal, and control to be returned to **bash**. The user may then manipulate the state of this job, using the **bg** command to continue it in the background, the **fg** command to continue it in the foreground, or the **kill** command to kill it. A **^Z** takes effect immediately, and has the additional side effect of causing pending output and typeahead to be discarded.

There are a number of ways to refer to a job in the shell. The character **%** introduces a job name. Job number *n* may be referred to as **%n**. A job may also be referred to using a prefix of the name used to start it, or using a substring that appears in its command line. For example, **%ce** refers to a stopped **ce** job. If a prefix matches more than one job, **bash** reports an error. Using **%?ce**, on the other hand, refers to any job containing the string **ce** in its command line. If the substring matches more than one job, **bash** reports an error. The symbols **%%** and **%+** refer to the shell's notion of the *current job*, which is the last job stopped while it was in the foreground or started in the background. The *previous job* may be referenced using **%-**. In output pertaining to jobs (e.g., the output of the **jobs** command), the current job is always flagged with a **+**, and the previous job with a **-**. A single **%** (with no accompanying job specification) also refers to the current job.

Simply naming a job can be used to bring it into the foreground: **%1** is a synonym for “**fg %1**”, bringing job 1 from the background into the foreground. Similarly, “**%1 &**” resumes job 1 in the background, equivalent to “**bg %1**”.

The shell learns immediately whenever a job changes state. Normally, **bash** waits until it is about to print a prompt before reporting changes in a job's status so as to not interrupt any other output. If the **-b** option to the **set** builtin command is enabled, **bash** reports such changes immediately. Any trap on **SIGCHLD** is executed for each child that exits.

If an attempt to exit **bash** is made while jobs are stopped, the shell prints a warning message. The **jobs** command may then be used to inspect their status. If a second attempt to exit is made without an intervening command, the shell does not print another warning, and the stopped jobs are terminated.

## PROMPTING

When executing interactively, **bash** displays the primary prompt **PS1** when it is ready to read a command, and the secondary prompt **PS2** when it needs more input to complete a command. **Bash** allows these prompt strings to be customized by inserting a number of backslash-escaped special characters that are decoded as follows:

|                   |                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>\a</b>         | an ASCII bell character (07)                                                                                                                                                                          |
| <b>\d</b>         | the date in "Weekday Month Date" format (e.g., "Tue May 26")                                                                                                                                          |
| <b>\D{format}</b> | the <i>format</i> is passed to <i>strftime(3)</i> and the result is inserted into the prompt string; an empty <i>format</i> results in a locale-specific time representation. The braces are required |
| <b>\e</b>         | an ASCII escape character (033)                                                                                                                                                                       |
| <b>\h</b>         | the hostname up to the first ‘.’                                                                                                                                                                      |
| <b>\H</b>         | the hostname                                                                                                                                                                                          |
| <b>\j</b>         | the number of jobs currently managed by the shell                                                                                                                                                     |
| <b>\l</b>         | the basename of the shell's terminal device name                                                                                                                                                      |
| <b>\n</b>         | newline                                                                                                                                                                                               |
| <b>\r</b>         | carriage return                                                                                                                                                                                       |
| <b>\s</b>         | the name of the shell, the basename of <b>\$0</b> (the portion following the final slash)                                                                                                             |
| <b>\t</b>         | the current time in 24-hour HH:MM:SS format                                                                                                                                                           |
| <b>\T</b>         | the current time in 12-hour HH:MM:SS format                                                                                                                                                           |
| <b>\@</b>         | the current time in 12-hour am/pm format                                                                                                                                                              |
| <b>\A</b>         | the current time in 24-hour HH:MM format                                                                                                                                                              |
| <b>\u</b>         | the username of the current user                                                                                                                                                                      |

|                   |                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| <code>\v</code>   | the version of <b>bash</b> (e.g., 2.00)                                                                               |
| <code>\V</code>   | the release of <b>bash</b> , version + patch level (e.g., 2.00.0)                                                     |
| <code>\w</code>   | the current working directory, with <b>\$HOME</b> abbreviated with a tilde                                            |
| <code>\W</code>   | the basename of the current working directory, with <b>\$HOME</b> abbreviated with a tilde                            |
| <code>!</code>    | the history number of this command                                                                                    |
| <code>#</code>    | the command number of this command                                                                                    |
| <code>\$</code>   | if the effective UID is 0, a #, otherwise a \$                                                                        |
| <code>\nnn</code> | the character corresponding to the octal number <i>nnn</i>                                                            |
| <code>\</code>    | a backslash                                                                                                           |
| <code>[</code>    | begin a sequence of non-printing characters, which could be used to embed a terminal control sequence into the prompt |
| <code>]</code>    | end a sequence of non-printing characters                                                                             |

The command number and the history number are usually different: the history number of a command is its position in the history list, which may include commands restored from the history file (see **HISTORY** below), while the command number is the position in the sequence of commands executed during the current shell session. After the string is decoded, it is expanded via parameter expansion, command substitution, arithmetic expansion, and quote removal, subject to the value of the **promptvars** shell option (see the description of the **shopt** command under **SHELL BUILTIN COMMANDS** below).

## READLINE

This is the library that handles reading input when using an interactive shell, unless the **--noediting** option is given at shell invocation. By default, the line editing commands are similar to those of emacs. A vi-style line editing interface is also available. To turn off line editing after the shell is running, use the **+o emacs** or **+o vi** options to the **set** builtin (see **SHELL BUILTIN COMMANDS** below).

### Readline Notation

In this section, the emacs-style notation is used to denote keystrokes. Control keys are denoted by *C-key*, e.g., *C-n* means Control-N. Similarly, *meta* keys are denoted by *M-key*, so *M-x* means Meta-X. (On keyboards without a *meta* key, *M-x* means ESC *x*, i.e., press the Escape key then the *x* key. This makes ESC the *meta prefix*. The combination *M-C-x* means ESC-Control-*x*, or press the Escape key then hold the Control key while pressing the *x* key.)

Readline commands may be given numeric *arguments*, which normally act as a repeat count. Sometimes, however, it is the sign of the argument that is significant. Passing a negative argument to a command that acts in the forward direction (e.g., **kill-line**) causes that command to act in a backward direction. Commands whose behavior with arguments deviates from this are noted below.

When a command is described as *killing* text, the text deleted is saved for possible future retrieval (*yanking*). The killed text is saved in a *kill ring*. Consecutive kills cause the text to be accumulated into one unit, which can be yanked all at once. Commands which do not kill text separate the chunks of text on the kill ring.

### Readline Initialization

Readline is customized by putting commands in an initialization file (the *inputrc* file). The name of this file is taken from the value of the **INPUTRC** environment variable. If that variable is unset, readline will read both */etc/inputrc* and *~/.inputrc*. When a program which uses the readline library starts up, the initialization file is read, and the key bindings and variables are set. There are only a few basic constructs allowed in the readline initialization file. Blank lines are ignored. Lines beginning with a **#** are comments. Lines beginning with a **\$** indicate conditional constructs. Other lines denote key bindings and variable settings.

The default key-bindings may be changed with an *inputrc* file. Other programs that use this library may add their own commands and bindings.

For example, placing

**M-Control-u**: universal-argument

or

**C-Meta-u**: universal-argument

into the *inputrc* would make **M-C-u** execute the readline command *universal-argument*.

The following symbolic character names are recognized: *RUBOUT*, *DEL*, *ESC*, *LFD*, *NEWLINE*, *RET*, *RETURN*, *SPC*, *SPACE*, and *TAB*.

In addition to command names, readline allows keys to be bound to a string that is inserted when the key is pressed (a *macro*).

### Readline Key Bindings

The syntax for controlling key bindings in the *inputrc* file is simple. All that is required is the name of the command or the text of a macro and a key sequence to which it should be bound. The name may be specified in one of two ways: as a symbolic key name, possibly with *Meta-* or *Control-* prefixes, or as a key sequence.

When using the form **keyname**:*function-name* or *macro*, **keyname** is the name of a key spelled out in English. For example:

```
Control-u: universal-argument
Meta-Rubout: backward-kill-word
Control-o: "> output"
```

In the above example, *C-u* is bound to the function **universal-argument**, *M-DEL* is bound to the function **backward-kill-word**, and *C-o* is bound to run the macro expressed on the right hand side (that is, to insert the text `> output` into the line).

In the second form, "**keyseq**":*function-name* or *macro*, **keyseq** differs from **keyname** above in that strings denoting an entire key sequence may be specified by placing the sequence within double quotes. Some GNU Emacs style key escapes can be used, as in the following example, but the symbolic character names are not recognized.

```
"\C-u": universal-argument
"\C-x\C-r": re-read-init-file
"\e[11~": "Function Key 1"
```

In this example, *C-u* is again bound to the function **universal-argument**. *C-x C-r* is bound to the function **re-read-init-file**, and *ESC [ 11 ~* is bound to insert the text `Function Key 1`.

The full set of GNU Emacs style escape sequences is

```
\C- control prefix
\M- meta prefix
\e an escape character
\\ backslash
\" literal "
\' literal '
```

In addition to the GNU Emacs style escape sequences, a second set of backslash escapes is available:

```
\a alert (bell)
\b backspace
\d delete
\f form feed
\n newline
\r carriage return
\t horizontal tab
\v vertical tab
\nnn the eight-bit character whose value is the octal value nnn (one to three digits)
\xHH the eight-bit character whose value is the hexadecimal value HH (one or two hex digits)
```

When entering the text of a macro, single or double quotes must be used to indicate a macro definition. Unquoted text is assumed to be a function name. In the macro body, the backslash escapes described above are expanded. Backslash will quote any other character in the macro text, including `"` and `'`.

**Bash** allows the current readline key bindings to be displayed or modified with the **bind** builtin command.

The editing mode may be switched during interactive use by using the **-o** option to the **set** builtin command (see **SHELL BUILTIN COMMANDS** below).

### Readline Variables

Readline has variables that can be used to further customize its behavior. A variable may be set in the *inputrc* file with a statement of the form

```
set variable-name value
```

Except where noted, readline variables can take the values **On** or **Off** (without regard to case). Unrecognized variable names are ignored. When a variable value is read, empty or null values, "on" (case-insensitive), and "1" are equivalent to **On**. All other values are equivalent to **Off**. The variables and their default values are:

#### **bell-style (audible)**

Controls what happens when readline wants to ring the terminal bell. If set to **none**, readline never rings the bell. If set to **visible**, readline uses a visible bell if one is available. If set to **audible**, readline attempts to ring the terminal's bell.

#### **bind-tty-special-chars (On)**

If set to **On**, readline attempts to bind the control characters treated specially by the kernel's terminal driver to their readline equivalents.

#### **comment-begin ("#")**

The string that is inserted when the readline **insert-comment** command is executed. This command is bound to **M-#** in emacs mode and to **#** in vi command mode.

#### **completion-ignore-case (Off)**

If set to **On**, readline performs filename matching and completion in a case-insensitive fashion.

#### **completion-query-items (100)**

This determines when the user is queried about viewing the number of possible completions generated by the **possible-completions** command. It may be set to any integer value greater than or equal to zero. If the number of possible completions is greater than or equal to the value of this variable, the user is asked whether or not he wishes to view them; otherwise they are simply listed on the terminal.

#### **convert-meta (On)**

If set to **On**, readline will convert characters with the eighth bit set to an ASCII key sequence by stripping the eighth bit and prefixing an escape character (in effect, using escape as the *meta prefix*).

#### **disable-completion (Off)**

If set to **On**, readline will inhibit word completion. Completion characters will be inserted into the line as if they had been mapped to **self-insert**.

#### **editing-mode (emacs)**

Controls whether readline begins with a set of key bindings similar to *emacs* or *vi*. **editing-mode** can be set to either **emacs** or **vi**.

#### **enable-keypad (Off)**

When set to **On**, readline will try to enable the application keypad when it is called. Some systems need this to enable the arrow keys.

#### **expand-tilde (Off)**

If set to **on**, tilde expansion is performed when readline attempts word completion.

#### **history-preserve-point (Off)**

If set to **on**, the history code attempts to place point at the same location on each history line retrieved with **previous-history** or **next-history**.

#### **horizontal-scroll-mode (Off)**

When set to **On**, makes readline use a single line for display, scrolling the input horizontally on a single screen line when it becomes longer than the screen width rather than wrapping to a new line.

#### **input-meta (Off)**

If set to **On**, readline will enable eight-bit input (that is, it will not strip the high bit from the characters it reads), regardless of what the terminal claims it can support. The name **meta-flag** is a

synonym for this variable.

**isearch-terminators (“C-[C-J”)**

The string of characters that should terminate an incremental search without subsequently executing the character as a command. If this variable has not been given a value, the characters *ESC* and *C-J* will terminate an incremental search.

**keymap (emacs)**

Set the current readline keymap. The set of valid keymap names is *emacs*, *emacs-standard*, *emacs-meta*, *emacs-ctlx*, *vi*, *vi-command*, and *vi-insert*. *vi* is equivalent to *vi-command*; *emacs* is equivalent to *emacs-standard*. The default value is *emacs*; the value of **editing-mode** also affects the default keymap.

**mark-directories (On)**

If set to **On**, completed directory names have a slash appended.

**mark-modified-lines (Off)**

If set to **On**, history lines that have been modified are displayed with a preceding asterisk (\*).

**mark-symlinked-directories (Off)**

If set to **On**, completed names which are symbolic links to directories have a slash appended (subject to the value of **mark-directories**).

**match-hidden-files (On)**

This variable, when set to **On**, causes readline to match files whose names begin with a *.* (hidden files) when performing filename completion, unless the leading *.* is supplied by the user in the filename to be completed.

**output-meta (Off)**

If set to **On**, readline will display characters with the eighth bit set directly rather than as a meta-prefixed escape sequence.

**page-completions (On)**

If set to **On**, readline uses an internal *more*-like pager to display a screenful of possible completions at a time.

**print-completions-horizontally (Off)**

If set to **On**, readline will display completions with matches sorted horizontally in alphabetical order, rather than down the screen.

**show-all-if-ambiguous (Off)**

This alters the default behavior of the completion functions. If set to **on**, words which have more than one possible completion cause the matches to be listed immediately instead of ringing the bell.

**show-all-if-unmodified (Off)**

This alters the default behavior of the completion functions in a fashion similar to **show-all-if-ambiguous**. If set to **on**, words which have more than one possible completion without any possible partial completion (the possible completions don't share a common prefix) cause the matches to be listed immediately instead of ringing the bell.

**visible-stats (Off)**

If set to **On**, a character denoting a file's type as reported by *stat(2)* is appended to the filename when listing possible completions.

### Readline Conditional Constructs

Readline implements a facility similar in spirit to the conditional compilation features of the C preprocessor which allows key bindings and variable settings to be performed as the result of tests. There are four parser directives used.

**\$if** The **\$if** construct allows bindings to be made based on the editing mode, the terminal being used, or the application using readline. The text of the test extends to the end of the line; no characters are required to isolate it.

**mode** The **mode=** form of the **\$if** directive is used to test whether readline is in emacs or vi mode. This may be used in conjunction with the **set keymap** command, for instance, to set bindings in the *emacs-standard* and *emacs-ctlx* keymaps only if readline is starting out in emacs mode.

**term** The **term=** form may be used to include terminal-specific key bindings, perhaps to bind the key sequences output by the terminal's function keys. The word on the right side of the = is tested against the both full name of the terminal and the portion of the terminal name before the first -. This allows *sun* to match both *sun* and *sun-cmd*, for instance.

#### **application**

The **application** construct is used to include application-specific settings. Each program using the readline library sets the *application name*, and an initialization file can test for a particular value. This could be used to bind key sequences to functions useful for a specific program. For instance, the following command adds a key sequence that quotes the current or previous word in Bash:

```
$if Bash
Quote the current or previous word
"C-xq": "\eb\\"\ef\"
$endif
```

**\$endif** This command, as seen in the previous example, terminates an **\$if** command.

**\$else** Commands in this branch of the **\$if** directive are executed if the test fails.

#### **\$include**

This directive takes a single filename as an argument and reads commands and bindings from that file. For example, the following directive would read */etc/inputrc*:

```
$include /etc/inputrc
```

### **Searching**

Readline provides commands for searching through the command history (see **HISTORY** below) for lines containing a specified string. There are two search modes: *incremental* and *non-incremental*.

Incremental searches begin before the user has finished typing the search string. As each character of the search string is typed, readline displays the next entry from the history matching the string typed so far. An incremental search requires only as many characters as needed to find the desired history entry. The characters present in the value of the **isearch-terminators** variable are used to terminate an incremental search. If that variable has not been assigned a value the Escape and Control-J characters will terminate an incremental search. Control-G will abort an incremental search and restore the original line. When the search is terminated, the history entry containing the search string becomes the current line.

To find other matching entries in the history list, type Control-S or Control-R as appropriate. This will search backward or forward in the history for the next entry matching the search string typed so far. Any other key sequence bound to a readline command will terminate the search and execute that command. For instance, a *newline* will terminate the search and accept the line, thereby executing the command from the history list.

Readline remembers the last incremental search string. If two Control-Rs are typed without any intervening characters defining a new search string, any remembered search string is used.

Non-incremental searches read the entire search string before starting to search for matching history lines. The search string may be typed by the user or be part of the contents of the current line.

### **Readline Command Names**

The following is a list of the names of the commands and the default key sequences to which they are bound. Command names without an accompanying key sequence are unbound by default. In the following descriptions, *point* refers to the current cursor position, and *mark* refers to a cursor position saved by the **set-mark** command. The text between the point and mark is referred to as the *region*.

### **Commands for Moving**

#### **beginning-of-line (C-a)**

Move to the start of the current line.



**end-of-line (C-e)**

Move to the end of the line.

**forward-char (C-f)**

Move forward a character.

**backward-char (C-b)**

Move back a character.

**forward-word (M-f)**

Move forward to the end of the next word. Words are composed of alphanumeric characters (letters and digits).

**backward-word (M-b)**

Move back to the start of the current or previous word. Words are composed of alphanumeric characters (letters and digits).

**clear-screen (C-l)**

Clear the screen leaving the current line at the top of the screen. With an argument, refresh the current line without clearing the screen.

**redraw-current-line**

Refresh the current line.

**Commands for Manipulating the History****accept-line (Newline, Return)**

Accept the line regardless of where the cursor is. If this line is non-empty, add it to the history list according to the state of the **HISTCONTROL** variable. If the line is a modified history line, then restore the history line to its original state.

**previous-history (C-p)**

Fetch the previous command from the history list, moving back in the list.

**next-history (C-n)**

Fetch the next command from the history list, moving forward in the list.

**beginning-of-history (M-<)**

Move to the first line in the history.

**end-of-history (M->)**

Move to the end of the input history, i.e., the line currently being entered.

**reverse-search-history (C-r)**

Search backward starting at the current line and moving ‘up’ through the history as necessary. This is an incremental search.

**forward-search-history (C-s)**

Search forward starting at the current line and moving ‘down’ through the history as necessary. This is an incremental search.

**non-incremental-reverse-search-history (M-p)**

Search backward through the history starting at the current line using a non-incremental search for a string supplied by the user.

**non-incremental-forward-search-history (M-n)**

Search forward through the history using a non-incremental search for a string supplied by the user.

**history-search-forward**

Search forward through the history for the string of characters between the start of the current line and the point. This is a non-incremental search.

**history-search-backward**

Search backward through the history for the string of characters between the start of the current line and the point. This is a non-incremental search.

**yank-nth-arg (M-C-y)**

Insert the first argument to the previous command (usually the second word on the previous line) at point. With an argument *n*, insert the *n*th word from the previous command (the words in the previous command begin with word 0). A negative argument inserts the *n*th word from the end of the previous command. Once the argument *n* is computed, the argument is extracted as if the “!*n*” history expansion had been specified.

**yank-last-arg (M-., M-\_)**

Insert the last argument to the previous command (the last word of the previous history entry). With an argument, behave exactly like **yank-nth-arg**. Successive calls to **yank-last-arg** move back through the history list, inserting the last argument of each line in turn. The history expansion facilities are used to extract the last argument, as if the "!"\$ history expansion had been specified.

**shell-expand-line (M-C-e)**

Expand the line as the shell does. This performs alias and history expansion as well as all of the shell word expansions. See **HISTORY EXPANSION** below for a description of history expansion.

**history-expand-line (M-^)**

Perform history expansion on the current line. See **HISTORY EXPANSION** below for a description of history expansion.

**magic-space**

Perform history expansion on the current line and insert a space. See **HISTORY EXPANSION** below for a description of history expansion.

**alias-expand-line**

Perform alias expansion on the current line. See **ALIASES** above for a description of alias expansion.

**history-and-alias-expand-line**

Perform history and alias expansion on the current line.

**insert-last-argument (M-., M-\_)**

A synonym for **yank-last-arg**.

**operate-and-get-next (C-o)**

Accept the current line for execution and fetch the next line relative to the current line from the history for editing. Any argument is ignored.

**edit-and-execute-command (C-xC-e)**

Invoke an editor on the current command line, and execute the result as shell commands. **Bash** attempts to invoke **\$FCEDIT**, **\$EDITOR**, and *emacs* as the editor, in that order.

**Commands for Changing Text****delete-char (C-d)**

Delete the character at point. If point is at the beginning of the line, there are no characters in the line, and the last character typed was not bound to **delete-char**, then return EOF.

**backward-delete-char (Rubout)**

Delete the character behind the cursor. When given a numeric argument, save the deleted text on the kill ring.

**forward-backward-delete-char**

Delete the character under the cursor, unless the cursor is at the end of the line, in which case the character behind the cursor is deleted.

**quoted-insert (C-q, C-v)**

Add the next character typed to the line verbatim. This is how to insert characters like C-q, for example.

**tab-insert (C-v TAB)**

Insert a tab character.

**self-insert (a, b, A, 1, !, ...)**

Insert the character typed.

**transpose-chars (C-t)**

Drag the character before point forward over the character at point, moving point forward as well. If point is at the end of the line, then this transposes the two characters before point. Negative arguments have no effect.

**transpose-words (M-t)**

Drag the word before point past the word after point, moving point over that word as well. If point is at the end of the line, this transposes the last two words on the line.

**upcase-word (M-u)**

Uppercase the current (or following) word. With a negative argument, uppercase the previous word, but do not move point.

**downcase-word (M-l)**

Lowercase the current (or following) word. With a negative argument, lowercase the previous word, but do not move point.

**capitalize-word (M-c)**

Capitalize the current (or following) word. With a negative argument, capitalize the previous word, but do not move point.

**overwrite-mode**

Toggle overwrite mode. With an explicit positive numeric argument, switches to overwrite mode. With an explicit non-positive numeric argument, switches to insert mode. This command affects only **emacs** mode; **vi** mode does overwrite differently. Each call to *readline()* starts in insert mode. In overwrite mode, characters bound to **self-insert** replace the text at point rather than pushing the text to the right. Characters bound to **backward-delete-char** replace the character before point with a space. By default, this command is unbound.

**Killing and Yanking****kill-line (C-k)**

Kill the text from point to the end of the line.

**backward-kill-line (C-x Rubout)**

Kill backward to the beginning of the line.

**unix-line-discard (C-u)**

Kill backward from point to the beginning of the line. The killed text is saved on the kill-ring.

**kill-whole-line**

Kill all characters on the current line, no matter where point is.

**kill-word (M-d)**

Kill from point to the end of the current word, or if between words, to the end of the next word. Word boundaries are the same as those used by **forward-word**.

**backward-kill-word (M-Rubout)**

Kill the word behind point. Word boundaries are the same as those used by **backward-word**.

**unix-word-rubout (C-w)**

Kill the word behind point, using white space as a word boundary. The killed text is saved on the kill-ring.

**unix-filename-rubout**

Kill the word behind point, using white space and the slash character as the word boundaries. The killed text is saved on the kill-ring.

**delete-horizontal-space (M-\)**

Delete all spaces and tabs around point.

**kill-region**

Kill the text in the current region.

**copy-region-as-kill**

Copy the text in the region to the kill buffer.

**copy-backward-word**

Copy the word before point to the kill buffer. The word boundaries are the same as **backward-word**.

**copy-forward-word**

Copy the word following point to the kill buffer. The word boundaries are the same as **forward-word**.

**yank (C-y)**

Yank the top of the kill ring into the buffer at point.

**yank-pop (M-y)**

Rotate the kill ring, and yank the new top. Only works following **yank** or **yank-pop**.

## Numeric Arguments

### **digit-argument** (M-0, M-1, ..., M--)

Add this digit to the argument already accumulating, or start a new argument. M-- starts a negative argument.

### **universal-argument**

This is another way to specify an argument. If this command is followed by one or more digits, optionally with a leading minus sign, those digits define the argument. If the command is followed by digits, executing **universal-argument** again ends the numeric argument, but is otherwise ignored. As a special case, if this command is immediately followed by a character that is neither a digit or minus sign, the argument count for the next command is multiplied by four. The argument count is initially one, so executing this function the first time makes the argument count four, a second time makes the argument count sixteen, and so on.

## Completing

### **complete** (TAB)

Attempt to perform completion on the text before point. **Bash** attempts completion treating the text as a variable (if the text begins with \$), username (if the text begins with ~), hostname (if the text begins with @), or command (including aliases and functions) in turn. If none of these produces a match, filename completion is attempted.

### **possible-completions** (M-?)

List the possible completions of the text before point.

### **insert-completions** (M-\*)

Insert all completions of the text before point that would have been generated by **possible-completions**.

### **menu-complete**

Similar to **complete**, but replaces the word to be completed with a single match from the list of possible completions. Repeated execution of **menu-complete** steps through the list of possible completions, inserting each match in turn. At the end of the list of completions, the bell is rung (subject to the setting of **bell-style**) and the original text is restored. An argument of *n* moves *n* positions forward in the list of matches; a negative argument may be used to move backward through the list. This command is intended to be bound to **TAB**, but is unbound by default.

### **delete-char-or-list**

Deletes the character under the cursor if not at the beginning or end of the line (like **delete-char**). If at the end of the line, behaves identically to **possible-completions**. This command is unbound by default.

### **complete-filename** (M-/)

Attempt filename completion on the text before point.

### **possible-filename-completions** (C-x /)

List the possible completions of the text before point, treating it as a filename.

### **complete-username** (M-~)

Attempt completion on the text before point, treating it as a username.

### **possible-username-completions** (C-x ~)

List the possible completions of the text before point, treating it as a username.

### **complete-variable** (M-\$)

Attempt completion on the text before point, treating it as a shell variable.

### **possible-variable-completions** (C-x \$)

List the possible completions of the text before point, treating it as a shell variable.

### **complete-hostname** (M-@)

Attempt completion on the text before point, treating it as a hostname.

### **possible-hostname-completions** (C-x @)

List the possible completions of the text before point, treating it as a hostname.

### **complete-command** (M-!)

Attempt completion on the text before point, treating it as a command name. Command completion attempts to match the text against aliases, reserved words, shell functions, shell builtins, and finally executable filenames, in that order.

**possible-command-completions (C-x !)**

List the possible completions of the text before point, treating it as a command name.

**dynamic-complete-history (M-TAB)**

Attempt completion on the text before point, comparing the text against lines from the history list for possible completion matches.

**complete-into-braces (M-{)**

Perform filename completion and insert the list of possible completions enclosed within braces so the list is available to the shell (see **Brace Expansion** above).

**Keyboard Macros****start-kbd-macro (C-x (**

Begin saving the characters typed into the current keyboard macro.

**end-kbd-macro (C-x )**

Stop saving the characters typed into the current keyboard macro and store the definition.

**call-last-kbd-macro (C-x e)**

Re-execute the last keyboard macro defined, by making the characters in the macro appear as if typed at the keyboard.

**Miscellaneous****re-read-init-file (C-x C-r)**

Read in the contents of the *inputrc* file, and incorporate any bindings or variable assignments found there.

**abort (C-g)**

Abort the current editing command and ring the terminal's bell (subject to the setting of **bell-style**).

**do-uppercase-version (M-a, M-b, M-x, ...)**

If the metaified character *x* is lowercase, run the command that is bound to the corresponding uppercase character.

**prefix-meta (ESC)**

Metafy the next character typed. **ESC f** is equivalent to **Meta-f**.

**undo (C-\_, C-x C-u)**

Incremental undo, separately remembered for each line.

**revert-line (M-r)**

Undo all changes made to this line. This is like executing the **undo** command enough times to return the line to its initial state.

**tilde-expand (M-&)**

Perform tilde expansion on the current word.

**set-mark (C-@, M-<space>)**

Set the mark to the point. If a numeric argument is supplied, the mark is set to that position.

**exchange-point-and-mark (C-x C-x)**

Swap the point with the mark. The current cursor position is set to the saved position, and the old cursor position is saved as the mark.

**character-search (C-])**

A character is read and point is moved to the next occurrence of that character. A negative count searches for previous occurrences.

**character-search-backward (M-C-])**

A character is read and point is moved to the previous occurrence of that character. A negative count searches for subsequent occurrences.

**insert-comment (M-#)**

Without a numeric argument, the value of the readline **comment-begin** variable is inserted at the beginning of the current line. If a numeric argument is supplied, this command acts as a toggle: if the characters at the beginning of the line do not match the value of **comment-begin**, the value is inserted, otherwise the characters in **comment-begin** are deleted from the beginning of the line. In either case, the line is accepted as if a newline had been typed. The default value of **comment-begin** causes this command to make the current line a shell comment. If a numeric argument causes the comment character to be removed, the line will be executed by the shell.

**glob-complete-word (M-g)**

The word before point is treated as a pattern for pathname expansion, with an asterisk implicitly appended. This pattern is used to generate a list of matching file names for possible completions.

**glob-expand-word (C-x \*)**

The word before point is treated as a pattern for pathname expansion, and the list of matching file names is inserted, replacing the word. If a numeric argument is supplied, an asterisk is appended before pathname expansion.

**glob-list-expansions (C-x g)**

The list of expansions that would have been generated by **glob-expand-word** is displayed, and the line is redrawn. If a numeric argument is supplied, an asterisk is appended before pathname expansion.

**dump-functions**

Print all of the functions and their key bindings to the readline output stream. If a numeric argument is supplied, the output is formatted in such a way that it can be made part of an *inputrc* file.

**dump-variables**

Print all of the settable readline variables and their values to the readline output stream. If a numeric argument is supplied, the output is formatted in such a way that it can be made part of an *inputrc* file.

**dump-macros**

Print all of the readline key sequences bound to macros and the strings they output. If a numeric argument is supplied, the output is formatted in such a way that it can be made part of an *inputrc* file.

**display-shell-version (C-x C-v)**

Display version information about the current instance of **bash**.

**Programmable Completion**

When word completion is attempted for an argument to a command for which a completion specification (a *compspec*) has been defined using the **complete** builtin (see **SHELL BUILTIN COMMANDS** below), the programmable completion facilities are invoked.

First, the command name is identified. If a *compspec* has been defined for that command, the *compspec* is used to generate the list of possible completions for the word. If the command word is a full pathname, a *compspec* for the full pathname is searched for first. If no *compspec* is found for the full pathname, an attempt is made to find a *compspec* for the portion following the final slash.

Once a *compspec* has been found, it is used to generate the list of matching words. If a *compspec* is not found, the default **bash** completion as described above under **Completing** is performed.

First, the actions specified by the *compspec* are used. Only matches which are prefixed by the word being completed are returned. When the **-f** or **-d** option is used for filename or directory name completion, the shell variable **FIGNORE** is used to filter the matches.

Any completions specified by a filename expansion pattern to the **-G** option are generated next. The words generated by the pattern need not match the word being completed. The **GLOBIGNORE** shell variable is not used to filter the matches, but the **FIGNORE** variable is used.

Next, the string specified as the argument to the **-W** option is considered. The string is first split using the characters in the **IFS** special variable as delimiters. Shell quoting is honored. Each word is then expanded using brace expansion, tilde expansion, parameter and variable expansion, command substitution, and arithmetic expansion, as described above under **EXPANSION**. The results are split using the rules described above under **Word Splitting**. The results of the expansion are prefix-matched against the word being completed, and the matching words become the possible completions.

After these matches have been generated, any shell function or command specified with the **-F** and **-C** options is invoked. When the command or function is invoked, the **COMP\_LINE** and **COMP\_POINT** variables are assigned values as described above under **Shell Variables**. If a shell function is being invoked, the **COMP\_WORDS** and **COMP\_CWORD** variables are also set. When the function or command is invoked, the first argument is the name of the command whose arguments are being completed, the second argument is the word being completed, and the third argument is the word preceding the word being

completed on the current command line. No filtering of the generated completions against the word being completed is performed; the function or command has complete freedom in generating the matches.

Any function specified with **-F** is invoked first. The function may use any of the shell facilities, including the **compgen** builtin described below, to generate the matches. It must put the possible completions in the **COMPREPLY** array variable.

Next, any command specified with the **-C** option is invoked in an environment equivalent to command substitution. It should print a list of completions, one per line, to the standard output. Backslash may be used to escape a newline, if necessary.

After all of the possible completions are generated, any filter specified with the **-X** option is applied to the list. The filter is a pattern as used for pathname expansion; a **&** in the pattern is replaced with the text of the word being completed. A literal **&** may be escaped with a backslash; the backslash is removed before attempting a match. Any completion that matches the pattern will be removed from the list. A leading **!** negates the pattern; in this case any completion not matching the pattern will be removed.

Finally, any prefix and suffix specified with the **-P** and **-S** options are added to each member of the completion list, and the result is returned to the readline completion code as the list of possible completions.

If the previously-applied actions do not generate any matches, and the **-o dirnames** option was supplied to **complete** when the compspec was defined, directory name completion is attempted.

If the **-o plusdirs** option was supplied to **complete** when the compspec was defined, directory name completion is attempted and any matches are added to the results of the other actions.

By default, if a compspec is found, whatever it generates is returned to the completion code as the full set of possible completions. The default **bash** completions are not attempted, and the readline default of filename completion is disabled. If the **-o bashdefault** option was supplied to **complete** when the compspec was defined, the **bash** default completions are attempted if the compspec generates no matches. If the **-o default** option was supplied to **complete** when the compspec was defined, readline's default completion will be performed if the compspec (and, if attempted, the default **bash** completions) generate no matches.

When a compspec indicates that directory name completion is desired, the programmable completion functions force readline to append a slash to completed names which are symbolic links to directories, subject to the value of the **mark-directories** readline variable, regardless of the setting of the **mark-symlinked-directories** readline variable.

## HISTORY

When the **-o history** option to the **set** builtin is enabled, the shell provides access to the *command history*, the list of commands previously typed. The value of the **HISTSIZE** variable is used as the number of commands to save in a history list. The text of the last **HISTSIZE** commands (default 500) is saved. The shell stores each command in the history list prior to parameter and variable expansion (see **EXPANSION** above) but after history expansion is performed, subject to the values of the shell variables **HISTIGNORE** and **HISTCONTROL**.

On startup, the history is initialized from the file named by the variable **HISTFILE** (default **~/.bash\_history**). The file named by the value of **HISTFILE** is truncated, if necessary, to contain no more than the number of lines specified by the value of **HISTFILESIZE**. When an interactive shell exits, the last **HISTSIZE** lines are copied from the history list to **\$HISTFILE**. If the **histappend** shell option is enabled (see the description of **shopt** under **SHELL BUILTIN COMMANDS** below), the lines are appended to the history file, otherwise the history file is overwritten. If **HISTFILE** is unset, or if the history file is unwritable, the history is not saved. After saving the history, the history file is truncated to contain no more than **HISTFILESIZE** lines. If **HISTFILESIZE** is not set, no truncation is performed.

The builtin command **fc** (see **SHELL BUILTIN COMMANDS** below) may be used to list or edit and re-execute a portion of the history list. The **history** builtin may be used to display or modify the history list and manipulate the history file. When using command-line editing, search commands are available in each editing mode that provide access to the history list.

The shell allows control over which commands are saved on the history list. The **HISTCONTROL** and **HISTIGNORE** variables may be set to cause the shell to save only a subset of the commands entered. The

**cmdhist** shell option, if enabled, causes the shell to attempt to save each line of a multi-line command in the same history entry, adding semicolons where necessary to preserve syntactic correctness. The **lithist** shell option causes the shell to save the command with embedded newlines instead of semicolons. See the description of the **shopt** builtin below under **SHELL BUILTIN COMMANDS** for information on setting and unsetting shell options.

## HISTORY EXPANSION

The shell supports a history expansion feature that is similar to the history expansion in **csh**. This section describes what syntax features are available. This feature is enabled by default for interactive shells, and can be disabled using the **+H** option to the **set** builtin command (see **SHELL BUILTIN COMMANDS** below). Non-interactive shells do not perform history expansion by default.

History expansions introduce words from the history list into the input stream, making it easy to repeat commands, insert the arguments to a previous command into the current input line, or fix errors in previous commands quickly.

History expansion is performed immediately after a complete line is read, before the shell breaks it into words. It takes place in two parts. The first is to determine which line from the history list to use during substitution. The second is to select portions of that line for inclusion into the current one. The line selected from the history is the *event*, and the portions of that line that are acted upon are *words*. Various *modifiers* are available to manipulate the selected words. The line is broken into words in the same fashion as when reading input, so that several *metacharacter*-separated words surrounded by quotes are considered one word. History expansions are introduced by the appearance of the history expansion character, which is **!** by default. Only backslash (**\**) and single quotes can quote the history expansion character.

Several characters inhibit history expansion if found immediately following the history expansion character, even if it is unquoted: space, tab, newline, carriage return, and **=**. If the **extglob** shell option is enabled, ( will also inhibit expansion.

Several shell options settable with the **shopt** builtin may be used to tailor the behavior of history expansion. If the **histverify** shell option is enabled (see the description of the **shopt** builtin), and **readline** is being used, history substitutions are not immediately passed to the shell parser. Instead, the expanded line is reloaded into the **readline** editing buffer for further modification. If **readline** is being used, and the **histreedit** shell option is enabled, a failed history substitution will be reloaded into the **readline** editing buffer for correction. The **-p** option to the **history** builtin command may be used to see what a history expansion will do before using it. The **-s** option to the **history** builtin may be used to add commands to the end of the history list without actually executing them, so that they are available for subsequent recall.

The shell allows control of the various characters used by the history expansion mechanism (see the description of **histchars** above under **Shell Variables**).

### Event Designators

An event designator is a reference to a command line entry in the history list.

- !** Start a history substitution, except when followed by a **blank**, newline, carriage return, **=** or ( (when the **extglob** shell option is enabled using the **shopt** builtin).
- !n** Refer to command line *n*.
- !-n** Refer to the current command line minus *n*.
- !!** Refer to the previous command. This is a synonym for **!-1**.
- !string** Refer to the most recent command starting with *string*.
- !?string[?]** Refer to the most recent command containing *string*. The trailing **?** may be omitted if *string* is followed immediately by a newline.
- ^string1^string2^** Quick substitution. Repeat the last command, replacing *string1* with *string2*. Equivalent to **“!!:s/string1/string2”** (see **Modifiers** below).
- !#** The entire command line typed so far.



### Word Designators

Word designators are used to select desired words from the event. A **:** separates the event specification from the word designator. It may be omitted if the word designator begins with a **^**, **\$**, **\***, **-**, or **%**. Words are numbered from the beginning of the line, with the first word being denoted by 0 (zero). Words are inserted into the current line separated by single spaces.

#### 0 (zero)

The zeroth word. For the shell, this is the command word.

*n* The *n*th word.

**^** The first argument. That is, word 1.

**\$** The last argument.

**%** The word matched by the most recent *'?string?'* search.

*x-y* A range of words; *'-y'* abbreviates *'0-y'*.

**\*** All of the words but the zeroth. This is a synonym for *'1-\$'*. It is not an error to use **\*** if there is just one word in the event; the empty string is returned in that case.

**x\*** Abbreviates *x-\$*.

**x-** Abbreviates *x-\$* like **x\***, but omits the last word.

If a word designator is supplied without an event specification, the previous command is used as the event.

### Modifiers

After the optional word designator, there may appear a sequence of one or more of the following modifiers, each preceded by a **:**.

**h** Remove a trailing file name component, leaving only the head.

**t** Remove all leading file name components, leaving the tail.

**r** Remove a trailing suffix of the form *.xxx*, leaving the basename.

**e** Remove all but the trailing suffix.

**p** Print the new command but do not execute it.

**q** Quote the substituted words, escaping further substitutions.

**x** Quote the substituted words as with **q**, but break into words at **blanks** and newlines.

**/oldnew/**

Substitute *new* for the first occurrence of *old* in the event line. Any delimiter can be used in place of */*. The final delimiter is optional if it is the last character of the event line. The delimiter may be quoted in *old* and *new* with a single backslash. If **&** appears in *new*, it is replaced by *old*. A single backslash will quote the **&**. If *old* is null, it is set to the last *old* substituted, or, if no previous history substitutions took place, the last *string* in a *!?string[?]* search.

**&** Repeat the previous substitution.

**g** Cause changes to be applied over the entire event line. This is used in conjunction with **'s'** (e.g., **'gsloldnew/')** or **':&'**. If used with **'s'**, any delimiter can be used in place of */*, and the final delimiter is optional if it is the last character of the event line. An **a** may be used as a synonym for **g**.

**G** Apply the following **'s'** modifier once to each word in the event line.

### SHELL BUILTIN COMMANDS

Unless otherwise noted, each builtin command documented in this section as accepting options preceded by **-** accepts **--** to signify the end of the options. For example, the **:**, **true**, **false**, and **test** builtins do not accept options.

**:** [*arguments*]

No effect; the command does nothing beyond expanding *arguments* and performing any specified redirections. A zero exit code is returned.

**.** *filename* [*arguments*]

**source** *filename* [*arguments*]

Read and execute commands from *filename* in the current shell environment and return the exit status of the last command executed from *filename*. If *filename* does not contain a slash, file names in **PATH** are used to find the directory containing *filename*. The file searched for in **PATH** need not be executable. When **bash** is not in *posix mode*, the current directory is searched if no

file is found in **PATH**. If the **sourcepath** option to the **shopt** builtin command is turned off, the **PATH** is not searched. If any *arguments* are supplied, they become the positional parameters when *filename* is executed. Otherwise the positional parameters are unchanged. The return status is the status of the last command exited within the script (0 if no commands are executed), and false if *filename* is not found or cannot be read.

**alias** [-p] [*name*[=*value*] ...]

**Alias** with no arguments or with the **-p** option prints the list of aliases in the form **alias name=value** on standard output. When arguments are supplied, an alias is defined for each *name* whose *value* is given. A trailing space in *value* causes the next word to be checked for alias substitution when the alias is expanded. For each *name* in the argument list for which no *value* is supplied, the name and value of the alias is printed. **Alias** returns true unless a *name* is given for which no alias has been defined.

**bg** [*jobspec* ...]

Resume each suspended job *jobspec* in the background, as if it had been started with **&**. If *jobspec* is not present, the shell's notion of the *current job* is used. **bg jobspec** returns 0 unless run when job control is disabled or, when run with job control enabled, any specified *jobspec* was not found or was started without job control.

**bind** [-m *keymap*] [-lpsvPSV]

**bind** [-m *keymap*] [-q *function*] [-u *function*] [-r *keyseq*]

**bind** [-m *keymap*] -f *filename*

**bind** [-m *keymap*] -x *keyseq:shell-command*

**bind** [-m *keymap*] *keyseq:function-name*

**bind** *readline-command*

Display current **readline** key and function bindings, bind a key sequence to a **readline** function or macro, or set a **readline** variable. Each non-option argument is a command as it would appear in *.inputrc*, but each binding or command must be passed as a separate argument; e.g., '"\C-x\C-r": re-read-init-file'. Options, if supplied, have the following meanings:

**-m keymap**

Use *keymap* as the keymap to be affected by the subsequent bindings. Acceptable *keymap* names are *emacs*, *emacs-standard*, *emacs-meta*, *emacs-ctlx*, *vi*, *vi-move*, *vi-command*, and *vi-insert*. *vi* is equivalent to *vi-command*; *emacs* is equivalent to *emacs-standard*.

**-l** List the names of all **readline** functions.

**-p** Display **readline** function names and bindings in such a way that they can be re-read.

**-P** List current **readline** function names and bindings.

**-v** Display **readline** variable names and values in such a way that they can be re-read.

**-V** List current **readline** variable names and values.

**-s** Display **readline** key sequences bound to macros and the strings they output in such a way that they can be re-read.

**-S** Display **readline** key sequences bound to macros and the strings they output.

**-f filename**

Read key bindings from *filename*.

**-q function**

Query about which keys invoke the named *function*.

**-u function**

Unbind all keys bound to the named *function*.

**-r keyseq**

Remove any current binding for *keyseq*.

**-x keyseq:shell-command**

Cause *shell-command* to be executed whenever *keyseq* is entered.

The return value is 0 unless an unrecognized option is given or an error occurred.

**break** [*n*]

Exit from within a **for**, **while**, **until**, or **select** loop. If *n* is specified, break *n* levels. *n* must be  $\geq 1$ . If *n* is greater than the number of enclosing loops, all enclosing loops are exited. The return value is 0 unless the shell is not executing a loop when **break** is executed.

**builtin** *shell-builtin* [*arguments*]

Execute the specified shell builtin, passing it *arguments*, and return its exit status. This is useful when defining a function whose name is the same as a shell builtin, retaining the functionality of the builtin within the function. The **cd** builtin is commonly redefined this way. The return status is false if *shell-builtin* is not a shell builtin command.

**cd** [-L|-P] [*dir*]

Change the current directory to *dir*. The variable **HOME** is the default *dir*. The variable **CDPATH** defines the search path for the directory containing *dir*. Alternative directory names in **CDPATH** are separated by a colon (:). A null directory name in **CDPATH** is the same as the current directory, i.e., ".". If *dir* begins with a slash (/), then **CDPATH** is not used. The **-P** option says to use the physical directory structure instead of following symbolic links (see also the **-P** option to the **set** builtin command); the **-L** option forces symbolic links to be followed. An argument of **-** is equivalent to **\$OLDPWD**. If a non-empty directory name from **CDPATH** is used, or if **-** is the first argument, and the directory change is successful, the absolute pathname of the new working directory is written to the standard output. The return value is true if the directory was successfully changed; false otherwise.

**caller** [*expr*]

Returns the context of any active subroutine call (a shell function or a script executed with the **.** or **source** builtins. Without *expr*, **caller** displays the line number and source filename of the current subroutine call. If a non-negative integer is supplied as *expr*, **caller** displays the line number, subroutine name, and source file corresponding to that position in the current execution call stack. This extra information may be used, for example, to print a stack trace. The current frame is frame 0. The return value is 0 unless the shell is not executing a subroutine call or *expr* does not correspond to a valid position in the call stack.

**command** [-pVv] *command* [*arg* ...]

Run *command* with *args* suppressing the normal shell function lookup. Only builtin commands or commands found in the **PATH** are executed. If the **-p** option is given, the search for *command* is performed using a default value for **PATH** that is guaranteed to find all of the standard utilities. If either the **-V** or **-v** option is supplied, a description of *command* is printed. The **-v** option causes a single word indicating the command or file name used to invoke *command* to be displayed; the **-V** option produces a more verbose description. If the **-V** or **-v** option is supplied, the exit status is 0 if *command* was found, and 1 if not. If neither option is supplied and an error occurred or *command* cannot be found, the exit status is 127. Otherwise, the exit status of the **command** builtin is the exit status of *command*.

**compgen** [*option*] [*word*]

Generate possible completion matches for *word* according to the *options*, which may be any option accepted by the **complete** builtin with the exception of **-p** and **-r**, and write the matches to the standard output. When using the **-F** or **-C** options, the various shell variables set by the programmable completion facilities, while available, will not have useful values.

The matches will be generated in the same way as if the programmable completion code had generated them directly from a completion specification with the same flags. If *word* is specified, only those completions matching *word* will be displayed.

The return value is true unless an invalid option is supplied, or no matches were generated.

**complete** [-*abcdefghijklmnopqrstuvwxyz*] [-o *comp-option*] [-A *action*] [-G *globpat*] [-W *wordlist*] [-P *prefix*] [-S *suffix*]

[-X *filterpat*] [-F *function*] [-C *command*] *name* [*name* ...]

**complete -pr** [*name* ...]

Specify how arguments to each *name* should be completed. If the **-p** option is supplied, or if no options are supplied, existing completion specifications are printed in a way that allows them to be reused as input. The **-r** option removes a completion specification for each *name*, or, if no *names* are supplied, all completion specifications.

The process of applying these completion specifications when word completion is attempted is described above under **Programmable Completion**.

Other options, if specified, have the following meanings. The arguments to the **-G**, **-W**, and **-X** options (and, if necessary, the **-P** and **-S** options) should be quoted to protect them from expansion before the **complete** builtin is invoked.

**-o** *comp-option*

The *comp-option* controls several aspects of the compspec's behavior beyond the simple generation of completions. *comp-option* may be one of:

**bashdefault**

Perform the rest of the default **bash** completions if the compspec generates no matches.

**default** Use readline's default filename completion if the compspec generates no matches.

**dirnames**

Perform directory name completion if the compspec generates no matches.

**filenames**

Tell readline that the compspec generates filenames, so it can perform any file-name-specific processing (like adding a slash to directory names or suppressing trailing spaces). Intended to be used with shell functions.

**nospace** Tell readline not to append a space (the default) to words completed at the end of the line.

**plusdirs** After any matches defined by the compspec are generated, directory name completion is attempted and any matches are added to the results of the other actions.

**-A** *action*

The *action* may be one of the following to generate a list of possible completions:

**alias** Alias names. May also be specified as **-a**.

**arrayvar**

Array variable names.

**binding** Readline key binding names.

**builtin** Names of shell builtin commands. May also be specified as **-b**.

**command**

Command names. May also be specified as **-c**.

**directory**

Directory names. May also be specified as **-d**.

**disabled**

Names of disabled shell builtins.

**enabled** Names of enabled shell builtins.

**export** Names of exported shell variables. May also be specified as **-e**.

**file** File names. May also be specified as **-f**.

**function**

Names of shell functions.

**group** Group names. May also be specified as **-g**.

**helptopic**

Help topics as accepted by the **help** builtin.

**hostname**

Hostnames, as taken from the file specified by the **HOSTFILE** shell variable.

**job** Job names, if job control is active. May also be specified as **-j**.

**keyword**

Shell reserved words. May also be specified as **-k**.

**running** Names of running jobs, if job control is active.

**service** Service names. May also be specified as **-s**.

**setopt** Valid arguments for the **-o** option to the **set** builtin.

**shopt** Shell option names as accepted by the **shopt** builtin.

**signal** Signal names.

**stopped** Names of stopped jobs, if job control is active.

**user** User names. May also be specified as **-u**.

**variable** Names of all shell variables. May also be specified as **-v**.

**-G globpat**

The filename expansion pattern *globpat* is expanded to generate the possible completions.

**-W wordlist**

The *wordlist* is split using the characters in the **IFS** special variable as delimiters, and each resultant word is expanded. The possible completions are the members of the resultant list which match the word being completed.

**-C command**

*command* is executed in a subshell environment, and its output is used as the possible completions.

**-F function**

The shell function *function* is executed in the current shell environment. When it finishes, the possible completions are retrieved from the value of the **COMPREPLY** array variable.

**-X filterpat**

*filterpat* is a pattern as used for filename expansion. It is applied to the list of possible completions generated by the preceding options and arguments, and each completion matching *filterpat* is removed from the list. A leading **!** in *filterpat* negates the pattern; in this case, any completion not matching *filterpat* is removed.

**-P prefix**

*prefix* is added at the beginning of each possible completion after all other options have been applied.

**-S suffix**

*suffix* is appended to each possible completion after all other options have been applied.

The return value is true unless an invalid option is supplied, an option other than **-p** or **-r** is supplied without a *name* argument, an attempt is made to remove a completion specification for a *name* for which no specification exists, or an error occurs adding a completion specification.

**continue** [*n*]

Resume the next iteration of the enclosing **for**, **while**, **until**, or **select** loop. If *n* is specified, resume at the *n*th enclosing loop. *n* must be  $\geq 1$ . If *n* is greater than the number of enclosing loops, the last enclosing loop (the “top-level” loop) is resumed. The return value is 0 unless the shell is not executing a loop when **continue** is executed.

**declare** [**-afFirtx**] [**-p**] [*name*[=*value*] ...]**typeset** [**-afFirtx**] [**-p**] [*name*[=*value*] ...]

Declare variables and/or give them attributes. If no *names* are given then display the values of variables. The **-p** option will display the attributes and values of each *name*. When **-p** is used, additional options are ignored. The **-F** option inhibits the display of function definitions; only the function name and attributes are printed. If the **extdebug** shell option is enabled using **shopt**, the source file name and line number where the function is defined are displayed as well. The **-F** option implies **-f**. The following options can be used to restrict output to variables with the specified attribute or to give variables attributes:

- a** Each *name* is an array variable (see **Arrays** above).
- f** Use function names only.
- i** The variable is treated as an integer; arithmetic evaluation (see **ARITHMETIC EVALUATION**) is performed when the variable is assigned a value.
- r** Make *names* readonly. These names cannot then be assigned values by subsequent assignment statements or unset.
- t** Give each *name* the *trace* attribute. Traced functions inherit the **DEBUG** and **RETURN** traps from the calling shell. The trace attribute has no special meaning for variables.
- x** Mark *names* for export to subsequent commands via the environment.

Using **+** instead of **-** turns off the attribute instead, with the exception that **+a** may not be used to destroy an array variable. When used in a function, makes each *name* local, as with the **local** command. If a variable name is followed by **=value**, the value of the variable is set to *value*. The return value is 0 unless an invalid option is encountered, an attempt is made to define a function using **-f foo=bar**, an attempt is made to assign a value to a readonly variable, an attempt is made to assign a value to an array variable without using the compound assignment syntax (see **Arrays** above), one of the *names* is not a valid shell variable name, an attempt is made to turn off readonly status for a readonly variable, an attempt is made to turn off array status for an array variable, or an attempt is made to display a non-existent function with **-f**.

#### **dirs [-clpv] [+n] [-n]**

Without options, displays the list of currently remembered directories. The default display is on a single line with directory names separated by spaces. Directories are added to the list with the **pushd** command; the **popd** command removes entries from the list.

- +n** Displays the *n*th entry counting from the left of the list shown by **dirs** when invoked without options, starting with zero.
- n** Displays the *n*th entry counting from the right of the list shown by **dirs** when invoked without options, starting with zero.
- c** Clears the directory stack by deleting all of the entries.
- l** Produces a longer listing; the default listing format uses a tilde to denote the home directory.
- p** Print the directory stack with one entry per line.
- v** Print the directory stack with one entry per line, prefixing each entry with its index in the stack.

The return value is 0 unless an invalid option is supplied or *n* indexes beyond the end of the directory stack.

#### **disown [-ar] [-h] [jobspec ...]**

Without options, each *jobspec* is removed from the table of active jobs. If the **-h** option is given, each *jobspec* is not removed from the table, but is marked so that **SIGHUP** is not sent to the job if the shell receives a **SIGHUP**. If no *jobspec* is present, and neither the **-a** nor the **-r** option is supplied, the *current job* is used. If no *jobspec* is supplied, the **-a** option means to remove or mark all jobs; the **-r** option without a *jobspec* argument restricts operation to running jobs. The return value is 0 unless a *jobspec* does not specify a valid job.

#### **echo [-neE] [arg ...]**

Output the *args*, separated by spaces, followed by a newline. The return status is always 0. If **-n** is specified, the trailing newline is suppressed. If the **-e** option is given, interpretation of the following backslash-escaped characters is enabled. The **-E** option disables the interpretation of these escape characters, even on systems where they are interpreted by default. The **xpg\_echo** shell option may be used to dynamically determine whether or not **echo** expands these escape characters by default. **echo** does not interpret **--** to mean the end of options. **echo** interprets the following escape sequences:

- \a** alert (bell)
- \b** backspace

|                    |                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------|
| <code>\c</code>    | suppress trailing newline                                                                      |
| <code>\e</code>    | an escape character                                                                            |
| <code>\f</code>    | form feed                                                                                      |
| <code>\n</code>    | new line                                                                                       |
| <code>\r</code>    | carriage return                                                                                |
| <code>\t</code>    | horizontal tab                                                                                 |
| <code>\v</code>    | vertical tab                                                                                   |
| <code>\\</code>    | backslash                                                                                      |
| <code>\0nnn</code> | the eight-bit character whose value is the octal value <i>nnn</i> (zero to three octal digits) |
| <code>\xHH</code>  | the eight-bit character whose value is the hexadecimal value <i>HH</i> (one or two hex digits) |

**enable** [**-adnps**] [**-f** *filename*] [*name* ...]

Enable and disable builtin shell commands. Disabling a builtin allows a disk command which has the same name as a shell builtin to be executed without specifying a full pathname, even though the shell normally searches for builtins before disk commands. If **-n** is used, each *name* is disabled; otherwise, *names* are enabled. For example, to use the **test** binary found via the **PATH** instead of the shell builtin version, run **enable -n test**. The **-f** option means to load the new builtin command *name* from shared object *filename*, on systems that support dynamic loading. The **-d** option will delete a builtin previously loaded with **-f**. If no *name* arguments are given, or if the **-p** option is supplied, a list of shell builtins is printed. With no other option arguments, the list consists of all enabled shell builtins. If **-n** is supplied, only disabled builtins are printed. If **-a** is supplied, the list printed includes all builtins, with an indication of whether or not each is enabled. If **-s** is supplied, the output is restricted to the POSIX *special* builtins. The return value is 0 unless a *name* is not a shell builtin or there is an error loading a new builtin from a shared object.

**eval** [*arg* ...]

The *args* are read and concatenated together into a single command. This command is then read and executed by the shell, and its exit status is returned as the value of **eval**. If there are no *args*, or only null arguments, **eval** returns 0.

**exec** [**-cl**] [**-a** *name*] [*command* [*arguments*]]

If *command* is specified, it replaces the shell. No new process is created. The *arguments* become the arguments to *command*. If the **-l** option is supplied, the shell places a dash at the beginning of the zeroth arg passed to *command*. This is what **login(1)** does. The **-c** option causes *command* to be executed with an empty environment. If **-a** is supplied, the shell passes *name* as the zeroth argument to the executed command. If *command* cannot be executed for some reason, a non-interactive shell exits, unless the shell option **execfail** is enabled, in which case it returns failure. An interactive shell returns failure if the file cannot be executed. If *command* is not specified, any redirections take effect in the current shell, and the return status is 0. If there is a redirection error, the return status is 1.

**exit** [*n*] Cause the shell to exit with a status of *n*. If *n* is omitted, the exit status is that of the last command executed. A trap on **EXIT** is executed before the shell terminates.

**export** [**-fn**] [*name*[=*word*]] ...

**export -p**

The supplied *names* are marked for automatic export to the environment of subsequently executed commands. If the **-f** option is given, the *names* refer to functions. If no *names* are given, or if the **-p** option is supplied, a list of all names that are exported in this shell is printed. The **-n** option causes the export property to be removed from each *name*. If a variable name is followed by *=word*, the value of the variable is set to *word*. **export** returns an exit status of 0 unless an invalid option is encountered, one of the *names* is not a valid shell variable name, or **-f** is supplied with a *name* that is not a function.

**fc** [**-e** *ename*] [**-nlr**] [*first*] [*last*]

**fc** **-s** [*pat=rep*] [*cmd*]

Fix Command. In the first form, a range of commands from *first* to *last* is selected from the history list. *First* and *last* may be specified as a string (to locate the last command beginning with that string) or as a number (an index into the history list, where a negative number is used as an offset from the current command number). If *last* is not specified it is set to the current command for listing (so that `fc -l -10` prints the last 10 commands) and to *first* otherwise. If *first* is not specified it is set to the previous command for editing and `-16` for listing.

The **-n** option suppresses the command numbers when listing. The **-r** option reverses the order of the commands. If the **-l** option is given, the commands are listed on standard output. Otherwise, the editor given by *ename* is invoked on a file containing those commands. If *ename* is not given, the value of the **FCEDIT** variable is used, and the value of **EDITOR** if **FCEDIT** is not set. If neither variable is set, *vi* is used. When editing is complete, the edited commands are echoed and executed.

In the second form, *command* is re-executed after each instance of *pat* is replaced by *rep*. A useful alias to use with this is `r='fc -s'`, so that typing `r cc` runs the last command beginning with `cc` and typing `r` re-executes the last command.

If the first form is used, the return value is 0 unless an invalid option is encountered or *first* or *last* specify history lines out of range. If the **-e** option is supplied, the return value is the value of the last command executed or failure if an error occurs with the temporary file of commands. If the second form is used, the return status is that of the command re-executed, unless *cmd* does not specify a valid history line, in which case **fc** returns failure.

**fg** [*jobspec*]

Resume *jobspec* in the foreground, and make it the current job. If *jobspec* is not present, the shell's notion of the *current job* is used. The return value is that of the command placed into the foreground, or failure if run when job control is disabled or, when run with job control enabled, if *jobspec* does not specify a valid job or *jobspec* specifies a job that was started without job control.

**getopts** *optstring name* [*args*]

**getopts** is used by shell procedures to parse positional parameters. *optstring* contains the option characters to be recognized; if a character is followed by a colon, the option is expected to have an argument, which should be separated from it by white space. The colon and question mark characters may not be used as option characters. Each time it is invoked, **getopts** places the next option in the shell variable *name*, initializing *name* if it does not exist, and the index of the next argument to be processed into the variable **OPTIND**. **OPTIND** is initialized to 1 each time the shell or a shell script is invoked. When an option requires an argument, **getopts** places that argument into the variable **OPTARG**. The shell does not reset **OPTIND** automatically; it must be manually reset between multiple calls to **getopts** within the same shell invocation if a new set of parameters is to be used.

When the end of options is encountered, **getopts** exits with a return value greater than zero. **OPTIND** is set to the index of the first non-option argument, and **name** is set to `?`.

**getopts** normally parses the positional parameters, but if more arguments are given in *args*, **getopts** parses those instead.

**getopts** can report errors in two ways. If the first character of *optstring* is a colon, *silent* error reporting is used. In normal operation diagnostic messages are printed when invalid options or missing option arguments are encountered. If the variable **OPTERR** is set to 0, no error messages will be displayed, even if the first character of *optstring* is not a colon.

If an invalid option is seen, **getopts** places `?` into *name* and, if not silent, prints an error message and unsets **OPTARG**. If **getopts** is silent, the option character found is placed in **OPTARG** and no



diagnostic message is printed.

If a required argument is not found, and **getopts** is not silent, a question mark (?) is placed in *name*, **OPTARG** is unset, and a diagnostic message is printed. If **getopts** is silent, then a colon (:) is placed in *name* and **OPTARG** is set to the option character found.

**getopts** returns true if an option, specified or unspecified, is found. It returns false if the end of options is encountered or an error occurs.

**hash** [-lr] [-p *filename*] [-dt] [*name*]

For each *name*, the full file name of the command is determined by searching the directories in **\$PATH** and remembered. If the **-p** option is supplied, no path search is performed, and *filename* is used as the full file name of the command. The **-r** option causes the shell to forget all remembered locations. The **-d** option causes the shell to forget the remembered location of each *name*. If the **-t** option is supplied, the full pathname to which each *name* corresponds is printed. If multiple *name* arguments are supplied with **-t**, the *name* is printed before the hashed full pathname. The **-l** option causes output to be displayed in a format that may be reused as input. If no arguments are given, or if only **-l** is supplied, information about remembered commands is printed. The return status is true unless a *name* is not found or an invalid option is supplied.

**help** [-s] [*pattern*]

Display helpful information about builtin commands. If *pattern* is specified, **help** gives detailed help on all commands matching *pattern*; otherwise help for all the builtins and shell control structures is printed. The **-s** option restricts the information displayed to a short usage synopsis. The return status is 0 unless no command matches *pattern*.

**history** [*n*]

**history** -c

**history** -d *offset*

**history** -anrw [*filename*]

**history** -p *arg* [*arg* ...]

**history** -s *arg* [*arg* ...]

With no options, display the command history list with line numbers. Lines listed with a \* have been modified. An argument of *n* lists only the last *n* lines. If the shell variable **HISTTIMEFORMAT** is set and not null, it is used as a format string for *strftime*(3) to display the time stamp associated with each displayed history entry. No intervening blank is printed between the formatted time stamp and the history line. If *filename* is supplied, it is used as the name of the history file; if not, the value of **HISTFILE** is used. Options, if supplied, have the following meanings:

**-c** Clear the history list by deleting all the entries.

**-d** *offset*

Delete the history entry at position *offset*.

**-a** Append the “new” history lines (history lines entered since the beginning of the current **bash** session) to the history file.

**-n** Read the history lines not already read from the history file into the current history list. These are lines appended to the history file since the beginning of the current **bash** session.

**-r** Read the contents of the history file and use them as the current history.

**-w** Write the current history to the history file, overwriting the history file’s contents.

**-p** Perform history substitution on the following *args* and display the result on the standard output. Does not store the results in the history list. Each *arg* must be quoted to disable normal history expansion.

**-s** Store the *args* in the history list as a single entry. The last command in the history list is removed before the *args* are added.

If the **HISTTIMEFORMAT** is set, the time stamp information associated with each history entry is written to the history file. The return value is 0 unless an invalid option is encountered, an error occurs while reading or writing the history file, an invalid *offset* is supplied as an argument to **-d**,

or the history expansion supplied as an argument to **-p** fails.

**jobs** [**-lnprs**] [*jobspec* ...]

**jobs** **-x** *command* [*args* ...]

The first form lists the active jobs. The options have the following meanings:

- l** List process IDs in addition to the normal information.
- p** List only the process ID of the job's process group leader.
- n** Display information only about jobs that have changed status since the user was last notified of their status.
- r** Restrict output to running jobs.
- s** Restrict output to stopped jobs.

If *jobspec* is given, output is restricted to information about that job. The return status is 0 unless an invalid option is encountered or an invalid *jobspec* is supplied.

If the **-x** option is supplied, **jobs** replaces any *jobspec* found in *command* or *args* with the corresponding process group ID, and executes *command* passing it *args*, returning its exit status.

**kill** [**-s** *sigspec* | **-n** *signum* | **-sigspec**] [*pid* | *jobspec*] ...

**kill** **-l** [*sigspec* | *exit\_status*]

Send the signal named by *sigspec* or *signum* to the processes named by *pid* or *jobspec*. *sigspec* is either a case-insensitive signal name such as **SIGKILL** (with or without the **SIG** prefix) or a signal number; *signum* is a signal number. If *sigspec* is not present, then **SIGTERM** is assumed. An argument of **-l** lists the signal names. If any arguments are supplied when **-l** is given, the names of the signals corresponding to the arguments are listed, and the return status is 0. The *exit\_status* argument to **-l** is a number specifying either a signal number or the exit status of a process terminated by a signal. **kill** returns true if at least one signal was successfully sent, or false if an error occurs or an invalid option is encountered.

**let** *arg* [*arg* ...]

Each *arg* is an arithmetic expression to be evaluated (see **ARITHMETIC EVALUATION**). If the last *arg* evaluates to 0, **let** returns 1; 0 is returned otherwise.

**local** [*option*] [*name*[=*value*] ...]

For each argument, a local variable named *name* is created, and assigned *value*. The *option* can be any of the options accepted by **declare**. When **local** is used within a function, it causes the variable *name* to have a visible scope restricted to that function and its children. With no operands, **local** writes a list of local variables to the standard output. It is an error to use **local** when not within a function. The return status is 0 unless **local** is used outside a function, an invalid *name* is supplied, or *name* is a readonly variable.

**logout** Exit a login shell.

**popd** [**-n**] [**+n**] [**-n**]

Removes entries from the directory stack. With no arguments, removes the top directory from the stack, and performs a **cd** to the new top directory. Arguments, if supplied, have the following meanings:

- +n** Removes the *n*th entry counting from the left of the list shown by **dirs**, starting with zero. For example: **popd +0** removes the first directory, **popd +1** the second.
- n** Removes the *n*th entry counting from the right of the list shown by **dirs**, starting with zero. For example: **popd -0** removes the last directory, **popd -1** the next to last.
- n** Suppresses the normal change of directory when removing directories from the stack, so that only the stack is manipulated.

If the **popd** command is successful, a **dirs** is performed as well, and the return status is 0. **popd** returns false if an invalid option is encountered, the directory stack is empty, a non-existent directory stack entry is specified, or the directory change fails.

**printf** [-v *var*] *format* [*arguments*]

Write the formatted *arguments* to the standard output under the control of the *format*. The *format* is a character string which contains three types of objects: plain characters, which are simply copied to standard output, character escape sequences, which are converted and copied to the standard output, and format specifications, each of which causes printing of the next successive *argument*. In addition to the standard *printf*(1) formats, **%b** causes **printf** to expand backslash escape sequences in the corresponding *argument* (except that **\c** terminates output, backslashes in **\'**, **\''**, and **\?** are not removed, and octal escapes beginning with **\0** may contain up to four digits), and **%q** causes **printf** to output the corresponding *argument* in a format that can be reused as shell input.

The **-v** option causes the output to be assigned to the variable *var* rather than being printed to the standard output.

The *format* is reused as necessary to consume all of the *arguments*. If the *format* requires more *arguments* than are supplied, the extra format specifications behave as if a zero value or null string, as appropriate, had been supplied. The return value is zero on success, non-zero on failure.

**pushd** [-n] [*dir*]**pushd** [-n] [+n] [-n]

Adds a directory to the top of the directory stack, or rotates the stack, making the new top of the stack the current working directory. With no arguments, exchanges the top two directories and returns 0, unless the directory stack is empty. Arguments, if supplied, have the following meanings:

- +n** Rotates the stack so that the *n*th directory (counting from the left of the list shown by **dirs**, starting with zero) is at the top.
- n** Rotates the stack so that the *n*th directory (counting from the right of the list shown by **dirs**, starting with zero) is at the top.
- n** Suppresses the normal change of directory when adding directories to the stack, so that only the stack is manipulated.
- dir* Adds *dir* to the directory stack at the top, making it the new current working directory.

If the **pushd** command is successful, a **dirs** is performed as well. If the first form is used, **pushd** returns 0 unless the **cd** to *dir* fails. With the second form, **pushd** returns 0 unless the directory stack is empty, a non-existent directory stack element is specified, or the directory change to the specified new current directory fails.

**pwd** [-LP]

Print the absolute pathname of the current working directory. The pathname printed contains no symbolic links if the **-P** option is supplied or the **-o physical** option to the **set** builtin command is enabled. If the **-L** option is used, the pathname printed may contain symbolic links. The return status is 0 unless an error occurs while reading the name of the current directory or an invalid option is supplied.

**read** [-ers] [-u *fd*] [-t *timeout*] [-a *aname*] [-p *prompt*] [-n *nchars*] [-d *delim*] [*name* ...]

One line is read from the standard input, or from the file descriptor *fd* supplied as an argument to the **-u** option, and the first word is assigned to the first *name*, the second word to the second *name*, and so on, with leftover words and their intervening separators assigned to the last *name*. If there are fewer words read from the input stream than names, the remaining names are assigned empty values. The characters in **IFS** are used to split the line into words. The backslash character (**\**) may be used to remove any special meaning for the next character read and for line continuation. Options, if supplied, have the following meanings:

**-a aname**

The words are assigned to sequential indices of the array variable *aname*, starting at 0. *aname* is unset before any new values are assigned. Other *name* arguments are ignored.

**-d *delim***

The first character of *delim* is used to terminate the input line, rather than newline.

**-e**

If the standard input is coming from a terminal, **readline** (see **README** above) is used to obtain the line.

**-n *nchars***

**read** returns after reading *nchars* characters rather than waiting for a complete line of input.

**-p *prompt***

Display *prompt* on standard error, without a trailing newline, before attempting to read any input. The prompt is displayed only if input is coming from a terminal.

**-r**

Backslash does not act as an escape character. The backslash is considered to be part of the line. In particular, a backslash-newline pair may not be used as a line continuation.

**-s**

Silent mode. If input is coming from a terminal, characters are not echoed.

**-t *timeout***

Cause **read** to time out and return failure if a complete line of input is not read within *timeout* seconds. This option has no effect if **read** is not reading input from the terminal or a pipe.

**-u *fd***

Read input from file descriptor *fd*.

If no *names* are supplied, the line read is assigned to the variable **REPLY**. The return code is zero, unless end-of-file is encountered, **read** times out, or an invalid file descriptor is supplied as the argument to **-u**.

**readonly [-apf] [*name*[=*word*] ...]**

The given *names* are marked readonly; the values of these *names* may not be changed by subsequent assignment. If the **-f** option is supplied, the functions corresponding to the *names* are so marked. The **-a** option restricts the variables to arrays. If no *name* arguments are given, or if the **-p** option is supplied, a list of all readonly names is printed. The **-p** option causes output to be displayed in a format that may be reused as input. If a variable name is followed by *=word*, the value of the variable is set to *word*. The return status is 0 unless an invalid option is encountered, one of the *names* is not a valid shell variable name, or **-f** is supplied with a *name* that is not a function.

**return [*n*]**

Causes a function to exit with the return value specified by *n*. If *n* is omitted, the return status is that of the last command executed in the function body. If used outside a function, but during execution of a script by the **.** (**source**) command, it causes the shell to stop executing that script and return either *n* or the exit status of the last command executed within the script as the exit status of the script. If used outside a function and not during execution of a script by **.**, the return status is false. Any command associated with the **RETURN** trap is executed before execution resumes after the function or script.

**set [--abefhkmnptuvxBCHP] [--o *option*] [*arg* ...]**

Without options, the name and value of each shell variable are displayed in a format that can be reused as input for setting or resetting the currently-set variables. Read-only variables cannot be reset. In *posix mode*, only shell variables are listed. The output is sorted according to the current locale. When options are specified, they set or unset shell attributes. Any arguments remaining after the options are processed are treated as values for the positional parameters and are assigned, in order, to **\$1**, **\$2**, ... **\$n**. Options, if specified, have the following meanings:

**-a**

Automatically mark variables and functions which are modified or created for export to the environment of subsequent commands.

**-b**

Report the status of terminated background jobs immediately, rather than before the next primary prompt. This is effective only when job control is enabled.

**-e**

Exit immediately if a *simple command* (see **SHELL GRAMMAR** above) exits with a non-zero status. The shell does not exit if the command that fails is part of the command list immediately following a **while** or **until** keyword, part of the test in an *if* statement, part of a **&&** or **||** list, or if the command's return value is being inverted via **!**. A trap

- on **ERR**, if set, is executed before the shell exits.
- f** Disable pathname expansion.
  - h** Remember the location of commands as they are looked up for execution. This is enabled by default.
  - k** All arguments in the form of assignment statements are placed in the environment for a command, not just those that precede the command name.
  - m** Monitor mode. Job control is enabled. This option is on by default for interactive shells on systems that support it (see **JOB CONTROL** above). Background processes run in a separate process group and a line containing their exit status is printed upon their completion.
  - n** Read commands but do not execute them. This may be used to check a shell script for syntax errors. This is ignored by interactive shells.
  - o option-name**  
The *option-name* can be one of the following:
    - allexport** Same as **-a**.
    - braceexpand** Same as **-B**.
    - emacs** Use an emacs-style command line editing interface. This is enabled by default when the shell is interactive, unless the shell is started with the **--noediting** option.
    - errtrace** Same as **-E**.
    - functrace** Same as **-T**.
    - errexit** Same as **-e**.
    - hashall** Same as **-h**.
    - histexpand** Same as **-H**.
    - history** Enable command history, as described above under **HISTORY**. This option is on by default in interactive shells.
    - ignoreeof**  
The effect is as if the shell command `IGNOREEOF=10` had been executed (see **Shell Variables** above).
    - keyword** Same as **-k**.
    - monitor** Same as **-m**.
    - noclobber** Same as **-C**.
    - noexec** Same as **-n**.
    - noglob** Same as **-f**. **nolog** Currently ignored.
    - notify** Same as **-b**.
    - nounset** Same as **-u**.
    - onecmd** Same as **-t**.
    - physical** Same as **-P**.
    - pipefail** If set, the return value of a pipeline is the value of the last (rightmost) command to exit with a non-zero status, or zero if all commands in the pipeline exit successfully. This option is disabled by default.
    - posix** Change the behavior of **bash** where the default operation differs from the POSIX standard to match the standard (*posix mode*).
    - privileged** Same as **-p**.
    - verbose** Same as **-v**.
    - vi** Use a vi-style command line editing interface.

**xtrace** Same as **-x**.

If **-o** is supplied with no *option-name*, the values of the current options are printed. If **+o** is supplied with no *option-name*, a series of **set** commands to recreate the current option settings is displayed on the standard output.

- p** Turn on *privileged* mode. In this mode, the **\$ENV** and **\$BASH\_ENV** files are not processed, shell functions are not inherited from the environment, and the **SHELLOPTS** variable, if it appears in the environment, is ignored. If the shell is started with the effective user (group) id not equal to the real user (group) id, and the **-p** option is not supplied, these actions are taken and the effective user id is set to the real user id. If the **-p** option is supplied at startup, the effective user id is not reset. Turning this option off causes the effective user and group ids to be set to the real user and group ids.
- t** Exit after reading and executing one command.
- u** Treat unset variables as an error when performing parameter expansion. If expansion is attempted on an unset variable, the shell prints an error message, and, if not interactive, exits with a non-zero status.
- v** Print shell input lines as they are read.
- x** After expanding each *simple command*, **for** command, **case** command, **select** command, or arithmetic **for** command, display the expanded value of **PS4**, followed by the command and its expanded arguments or associated word list.
- B** The shell performs brace expansion (see **Brace Expansion** above). This is on by default.
- C** If set, **bash** does not overwrite an existing file with the **>**, **>&**, and **<>** redirection operators. This may be overridden when creating output files by using the redirection operator **>|** instead of **>**.
- E** If set, any trap on **ERR** is inherited by shell functions, command substitutions, and commands executed in a subshell environment. The **ERR** trap is normally not inherited in such cases.
- H** Enable **!** style history substitution. This option is on by default when the shell is interactive.
- P** If set, the shell does not follow symbolic links when executing commands such as **cd** that change the current working directory. It uses the physical directory structure instead. By default, **bash** follows the logical chain of directories when performing commands which change the current directory.
- T** If set, any traps on **DEBUG** and **RETURN** are inherited by shell functions, command substitutions, and commands executed in a subshell environment. The **DEBUG** and **RETURN** traps are normally not inherited in such cases.
- If no arguments follow this option, then the positional parameters are unset. Otherwise, the positional parameters are set to the *args*, even if some of them begin with a **-**.
- Signal the end of options, cause all remaining *args* to be assigned to the positional parameters. The **-x** and **-v** options are turned off. If there are no *args*, the positional parameters remain unchanged.

The options are off by default unless otherwise noted. Using **+** rather than **-** causes these options to be turned off. The options can also be specified as arguments to an invocation of the shell. The current set of options may be found in **\$-**. The return status is always true unless an invalid option is encountered.

#### **shift** [*n*]

The positional parameters from *n*+1 ... are renamed to **\$1** .... Parameters represented by the numbers **\$#** down to **\$#-n+1** are unset. *n* must be a non-negative number less than or equal to **\$#**. If *n* is 0, no parameters are changed. If *n* is not given, it is assumed to be 1. If *n* is greater than **\$#**, the positional parameters are not changed. The return status is greater than zero if *n* is greater than **\$#** or less than zero; otherwise 0.

**shopt** [-pqsu] [-o] [*optname* ...]

Toggle the values of variables controlling optional shell behavior. With no options, or with the **-p** option, a list of all settable options is displayed, with an indication of whether or not each is set. The **-p** option causes output to be displayed in a form that may be reused as input. Other options have the following meanings:

- s** Enable (set) each *optname*.
- u** Disable (unset) each *optname*.
- q** Suppresses normal output (quiet mode); the return status indicates whether the *optname* is set or unset. If multiple *optname* arguments are given with **-q**, the return status is zero if all *optnames* are enabled; non-zero otherwise.
- o** Restricts the values of *optname* to be those defined for the **-o** option to the **set** builtin.

If either **-s** or **-u** is used with no *optname* arguments, the display is limited to those options which are set or unset, respectively. Unless otherwise noted, the **shopt** options are disabled (unset) by default.

The return status when listing options is zero if all *optnames* are enabled, non-zero otherwise. When setting or unsetting options, the return status is zero unless an *optname* is not a valid shell option.

The list of **shopt** options is:

**cdable\_vars**

If set, an argument to the **cd** builtin command that is not a directory is assumed to be the name of a variable whose value is the directory to change to.

**cdspell** If set, minor errors in the spelling of a directory component in a **cd** command will be corrected. The errors checked for are transposed characters, a missing character, and one character too many. If a correction is found, the corrected file name is printed, and the command proceeds. This option is only used by interactive shells.

**checkhash**

If set, **bash** checks that a command found in the hash table exists before trying to execute it. If a hashed command no longer exists, a normal path search is performed.

**checkwinsize**

If set, **bash** checks the window size after each command and, if necessary, updates the values of **LINES** and **COLUMNS**.

**cmdhist** If set, **bash** attempts to save all lines of a multiple-line command in the same history entry. This allows easy re-editing of multi-line commands.

**compat31**

If set, **bash** changes its behavior to that of version 3.1 with respect to quoted arguments to the conditional command's **=~** operator.

**dotglob** If set, **bash** includes filenames beginning with a **.** in the results of pathname expansion.

**execfail** If set, a non-interactive shell will not exit if it cannot execute the file specified as an argument to the **exec** builtin command. An interactive shell does not exit if **exec** fails.

**expand\_aliases**

If set, aliases are expanded as described above under **ALIASES**. This option is enabled by default for interactive shells.

**extdebug**

If set, behavior intended for use by debuggers is enabled:

1. The **-F** option to the **declare** builtin displays the source file name and line number corresponding to each function name supplied as an argument.
2. If the command run by the **DEBUG** trap returns a non-zero value, the next command is skipped and not executed.
3. If the command run by the **DEBUG** trap returns a value of 2, and the shell is executing in a subroutine (a shell function or a shell script executed by the **.** or **source** builtins), a call to **return** is simulated.
4. **BASH\_ARGC** and **BASH\_ARGV** are updated as described in their descriptions above.

5. Function tracing is enabled: command substitution, shell functions, and subshells invoked with ( *command* ) inherit the **DEBUG** and **RETURN** traps.
  6. Error tracing is enabled: command substitution, shell functions, and subshells invoked with ( *command* ) inherit the **ERROR** trap.
- extglob** If set, the extended pattern matching features described above under **Pathname Expansion** are enabled.
- extquote** If set, `'string'` and `"string"` quoting is performed within `${parameter}` expansions enclosed in double quotes. This option is enabled by default.
- failglob** If set, patterns which fail to match filenames during pathname expansion result in an expansion error.
- force\_ignore** If set, the suffixes specified by the **FIGNORE** shell variable cause words to be ignored when performing word completion even if the ignored words are the only possible completions. See **SHELL VARIABLES** above for a description of **FIGNORE**. This option is enabled by default.
- gnu\_errfmt** If set, shell error messages are written in the standard GNU error message format.
- histappend** If set, the history list is appended to the file named by the value of the **HISTFILE** variable when the shell exits, rather than overwriting the file.
- histreedit** If set, and **readline** is being used, a user is given the opportunity to re-edit a failed history substitution.
- histverify** If set, and **readline** is being used, the results of history substitution are not immediately passed to the shell parser. Instead, the resulting line is loaded into the **readline** editing buffer, allowing further modification.
- hostcomplete** If set, and **readline** is being used, **bash** will attempt to perform hostname completion when a word containing a `@` is being completed (see **Completing** under **READLINE** above). This is enabled by default.
- huponexit** If set, **bash** will send **SIGHUP** to all jobs when an interactive login shell exits.
- interactive\_comments** If set, allow a word beginning with `#` to cause that word and all remaining characters on that line to be ignored in an interactive shell (see **COMMENTS** above). This option is enabled by default.
- lithist** If set, and the **cmdhist** option is enabled, multi-line commands are saved to the history with embedded newlines rather than using semicolon separators where possible.
- login\_shell** The shell sets this option if it is started as a login shell (see **INVOCATION** above). The value may not be changed.
- mailwarn** If set, and a file that **bash** is checking for mail has been accessed since the last time it was checked, the message “The mail in *mailfile* has been read” is displayed.
- no\_empty\_cmd\_completion** If set, and **readline** is being used, **bash** will not attempt to search the **PATH** for possible completions when completion is attempted on an empty line.
- nocaseglob** If set, **bash** matches filenames in a case-insensitive fashion when performing pathname expansion (see **Pathname Expansion** above).



**nocasematch**

If set, **bash** matches patterns in a case-insensitive fashion when performing matching while executing **case** or **[[** conditional commands.

**nullglob**

If set, **bash** allows patterns which match no files (see **Pathname Expansion** above) to expand to a null string, rather than themselves.

**progcomp**

If set, the programmable completion facilities (see **Programmable Completion** above) are enabled. This option is enabled by default.

**promptvars**

If set, prompt strings undergo parameter expansion, command substitution, arithmetic expansion, and quote removal after being expanded as described in **PROMPTING** above. This option is enabled by default.

**restricted\_shell**

The shell sets this option if it is started in restricted mode (see **RESTRICTED SHELL** below). The value may not be changed. This is not reset when the startup files are executed, allowing the startup files to discover whether or not a shell is restricted.

**shift\_verbose**

If set, the **shift** builtin prints an error message when the shift count exceeds the number of positional parameters.

**sourcepath**

If set, the **source** (.) builtin uses the value of **PATH** to find the directory containing the file supplied as an argument. This option is enabled by default.

**xpg\_echo**

If set, the **echo** builtin expands backslash-escape sequences by default.

**suspend** [-f]

Suspend the execution of this shell until it receives a **SIGCONT** signal. The **-f** option says not to complain if this is a login shell; just suspend anyway. The return status is 0 unless the shell is a login shell and **-f** is not supplied, or if job control is not enabled.

**test** *expr*

[ *expr* ] Return a status of 0 or 1 depending on the evaluation of the conditional expression *expr*. Each operator and operand must be a separate argument. Expressions are composed of the primaries described above under **CONDITIONAL EXPRESSIONS**. **test** does not accept any options, nor does it accept and ignore an argument of **--** as signifying the end of options.

Expressions may be combined using the following operators, listed in decreasing order of precedence.

**! *expr*** True if *expr* is false.

**( *expr* )** Returns the value of *expr*. This may be used to override the normal precedence of operators.

***expr1* -a *expr2***

True if both *expr1* and *expr2* are true.

***expr1* -o *expr2***

True if either *expr1* or *expr2* is true.

**test** and [ evaluate conditional expressions using a set of rules based on the number of arguments.

0 arguments

The expression is false.

1 argument

The expression is true if and only if the argument is not null.

2 arguments

If the first argument is **!**, the expression is true if and only if the second argument is null. If the first argument is one of the unary conditional operators listed above under **CONDITIONAL EXPRESSIONS**, the expression is true if the unary test is true. If the first argument is not a valid unary conditional operator, the expression is false.

## 3 arguments

If the second argument is one of the binary conditional operators listed above under **CONDITIONAL EXPRESSIONS**, the result of the expression is the result of the binary test using the first and third arguments as operands. If the first argument is **!**, the value is the negation of the two-argument test using the second and third arguments. If the first argument is exactly **(** and the third argument is exactly **)**, the result is the one-argument test of the second argument. Otherwise, the expression is false. The **-a** and **-o** operators are considered binary operators in this case.

## 4 arguments

If the first argument is **!**, the result is the negation of the three-argument expression composed of the remaining arguments. Otherwise, the expression is parsed and evaluated according to precedence using the rules listed above.

## 5 or more arguments

The expression is parsed and evaluated according to precedence using the rules listed above.

**times** Print the accumulated user and system times for the shell and for processes run from the shell. The return status is 0.

**trap** **[-lp]** **[[arg] sigspec ...]**

The command *arg* is to be read and executed when the shell receives signal(s) *sigspec*. If *arg* is absent (and there is a single *sigspec*) or **-**, each specified signal is reset to its original disposition (the value it had upon entrance to the shell). If *arg* is the null string the signal specified by each *sigspec* is ignored by the shell and by the commands it invokes. If *arg* is not present and **-p** has been supplied, then the trap commands associated with each *sigspec* are displayed. If no arguments are supplied or if only **-p** is given, **trap** prints the list of commands associated with each signal. The **-l** option causes the shell to print a list of signal names and their corresponding numbers. Each *sigspec* is either a signal name defined in *<signal.h>*, or a signal number. Signal names are case insensitive and the SIG prefix is optional. If a *sigspec* is **EXIT** (0) the command *arg* is executed on exit from the shell. If a *sigspec* is **DEBUG**, the command *arg* is executed before every *simple command*, *for* command, *case* command, *select* command, every arithmetic *for* command, and before the first command executes in a shell function (see **SHELL GRAMMAR** above). Refer to the description of the **extdebug** option to the **shopt** builtin for details of its effect on the **DEBUG** trap. If a *sigspec* is **ERR**, the command *arg* is executed whenever a simple command has a non-zero exit status, subject to the following conditions. The **ERR** trap is not executed if the failed command is part of the command list immediately following a **while** or **until** keyword, part of the test in an *if* statement, part of a **&&** or **||** list, or if the command's return value is being inverted via **!**. These are the same conditions obeyed by the **errexit** option. If a *sigspec* is **RETURN**, the command *arg* is executed each time a shell function or a script executed with the **.** or **source** builtins finishes executing. Signals ignored upon entry to the shell cannot be trapped or reset. Trapped signals that are not being ignored are reset to their original values in a child process when it is created. The return status is false if any *sigspec* is invalid; otherwise **trap** returns true.

**type** **[-aftpP]** *name* [*name* ...]

With no options, indicate how each *name* would be interpreted if used as a command name. If the **-t** option is used, **type** prints a string which is one of *alias*, *keyword*, *function*, *builtin*, or *file* if *name* is an alias, shell reserved word, function, builtin, or disk file, respectively. If the *name* is not found, then nothing is printed, and an exit status of false is returned. If the **-p** option is used, **type** either returns the name of the disk file that would be executed if *name* were specified as a command name, or nothing if `type -t name` would not return *file*. The **-P** option forces a **PATH** search for each *name*, even if `type -t name` would not return *file*. If a command is hashed, **-p** and **-P** print the hashed value, not necessarily the file that appears first in **PATH**. If the **-a** option is used, **type** prints all of the places that contain an executable named *name*. This includes aliases and functions, if and only if the **-p** option is not also used. The table of hashed commands is not consulted when using **-a**. The **-f** option suppresses shell function lookup, as with the **command**

builtin. **type** returns true if any of the arguments are found, false if none are found.

**ulimit** [**-SHacdefilmnpqrstuvx** [*limit*]]

Provides control over the resources available to the shell and to processes started by it, on systems that allow such control. The **-H** and **-S** options specify that the hard or soft limit is set for the given resource. A hard limit cannot be increased once it is set; a soft limit may be increased up to the value of the hard limit. If neither **-H** nor **-S** is specified, both the soft and hard limits are set. The value of *limit* can be a number in the unit specified for the resource or one of the special values **hard**, **soft**, or **unlimited**, which stand for the current hard limit, the current soft limit, and no limit, respectively. If *limit* is omitted, the current value of the soft limit of the resource is printed, unless the **-H** option is given. When more than one resource is specified, the limit name and unit are printed before the value. Other options are interpreted as follows:

- a** All current limits are reported
- c** The maximum size of core files created
- d** The maximum size of a process's data segment
- e** The maximum scheduling priority ("nice")
- f** The maximum size of files written by the shell and its children
- i** The maximum number of pending signals
- l** The maximum size that may be locked into memory
- m** The maximum resident set size
- n** The maximum number of open file descriptors (most systems do not allow this value to be set)
- p** The pipe size in 512-byte blocks (this may not be set)
- q** The maximum number of bytes in POSIX message queues
- r** The maximum real-time scheduling priority
- s** The maximum stack size
- t** The maximum amount of cpu time in seconds
- u** The maximum number of processes available to a single user
- v** The maximum amount of virtual memory available to the shell
- x** The maximum number of file locks

If *limit* is given, it is the new value of the specified resource (the **-a** option is display only). If no option is given, then **-f** is assumed. Values are in 1024-byte increments, except for **-t**, which is in seconds, **-p**, which is in units of 512-byte blocks, and **-n** and **-u**, which are unscaled values. The return status is 0 unless an invalid option or argument is supplied, or an error occurs while setting a new limit.

**umask** [**-p**] [**-S**] [*mode*]

The user file-creation mask is set to *mode*. If *mode* begins with a digit, it is interpreted as an octal number; otherwise it is interpreted as a symbolic mode mask similar to that accepted by *chmod*(1). If *mode* is omitted, the current value of the mask is printed. The **-S** option causes the mask to be printed in symbolic form; the default output is an octal number. If the **-p** option is supplied, and *mode* is omitted, the output is in a form that may be reused as input. The return status is 0 if the mode was successfully changed or if no *mode* argument was supplied, and false otherwise.

**unalias** [**-a**] [*name* ...]

Remove each *name* from the list of defined aliases. If **-a** is supplied, all alias definitions are removed. The return value is true unless a supplied *name* is not a defined alias.

**unset** [**-fv**] [*name* ...]

For each *name*, remove the corresponding variable or function. If no options are supplied, or the **-v** option is given, each *name* refers to a shell variable. Read-only variables may not be unset. If **-f** is specified, each *name* refers to a shell function, and the function definition is removed. Each unset variable or function is removed from the environment passed to subsequent commands. If any of **RANDOM**, **SECONDS**, **LINENO**, **HISTCMD**, **FUNCNAME**, **GROUPS**, or **DIRSTACK** are unset, they lose their special properties, even if they are subsequently reset. The exit status is true unless a *name* is readonly.

**wait** [*n* ...]

Wait for each specified process and return its termination status. Each *n* may be a process ID or a job specification; if a job spec is given, all processes in that job's pipeline are waited for. If *n* is not given, all currently active child processes are waited for, and the return status is zero. If *n* specifies a non-existent process or job, the return status is 127. Otherwise, the return status is the exit status of the last process or job waited for.

**RESTRICTED SHELL**

If **bash** is started with the name **rbash**, or the **-r** option is supplied at invocation, the shell becomes restricted. A restricted shell is used to set up an environment more controlled than the standard shell. It behaves identically to **bash** with the exception that the following are disallowed or not performed:

- changing directories with **cd**
- setting or unsetting the values of **SHELL**, **PATH**, **ENV**, or **BASH\_ENV**
- specifying command names containing /
- specifying a file name containing a / as an argument to the **.** builtin command
- Specifying a filename containing a slash as an argument to the **-p** option to the **hash** builtin command
- importing function definitions from the shell environment at startup
- parsing the value of **SHELLOPTS** from the shell environment at startup
- redirecting output using the **>**, **>|**, **<>**, **>&**, **&>**, and **>>** redirection operators
- using the **exec** builtin command to replace the shell with another command
- adding or deleting builtin commands with the **-f** and **-d** options to the **enable** builtin command
- Using the **enable** builtin command to enable disabled shell builtins
- specifying the **-p** option to the **command** builtin command
- turning off restricted mode with **set +r** or **set +o restricted**.

These restrictions are enforced after any startup files are read.

When a command that is found to be a shell script is executed (see **COMMAND EXECUTION** above), **rbash** turns off any restrictions in the shell spawned to execute the script.

**SEE ALSO**

*Bash Reference Manual*, Brian Fox and Chet Ramey  
*The Gnu Readline Library*, Brian Fox and Chet Ramey  
*The Gnu History Library*, Brian Fox and Chet Ramey  
*Portable Operating System Interface (POSIX) Part 2: Shell and Utilities*, IEEE  
*sh*(1), *ksh*(1), *csh*(1)  
*emacs*(1), *vi*(1)  
*readline*(3)

**FILES**

*/bin/bash*  
 The **bash** executable  
*/etc/profile*  
 The systemwide initialization file, executed for login shells  
*~/.bash\_profile*  
 The personal initialization file, executed for login shells  
*~/.bashrc*  
 The individual per-interactive-shell startup file  
*~/.bash\_logout*  
 The individual login shell cleanup file, executed when a login shell exits

*~/inputrc*

Individual *readline* initialization file

*/etc/inputrc*

System **readline** initialization file

## AUTHORS

Brian Fox, Free Software Foundation  
bfox@gnu.org

Chet Ramey, Case Western Reserve University  
chet@po.cwru.edu

## BUG REPORTS

If you find a bug in **bash**, you should report it. But first, you should make sure that it really is a bug, and that it appears in the latest version of **bash**. The latest version is always available from <ftp://ftp.gnu.org/pub/bash/>.

Once you have determined that a bug actually exists, use the *bashbug* command to submit a bug report. If you have a fix, you are encouraged to mail that as well! Suggestions and ‘philosophical’ bug reports may be mailed to [bug-bash@gnu.org](mailto:bug-bash@gnu.org) or posted to the Usenet newsgroup **gnu.bash.bug**.

ALL bug reports should include:

The version number of **bash**

The hardware and operating system

The compiler used to compile

A description of the bug behaviour

A short script or ‘recipe’ which exercises the bug

*bashbug* inserts the first three items automatically into the template it provides for filing a bug report.

Comments and bug reports concerning this manual page should be directed to [chet@po.cwru.edu](mailto:chet@po.cwru.edu).

## BUGS

It’s too big and too slow.

There are some subtle differences between **bash** and traditional versions of **sh**, mostly because of the **POSIX** specification.

Aliases are confusing in some uses.

Shell builtin commands and functions are not stoppable/restartable.

Compound commands and command sequences of the form ‘a ; b ; c’ are not handled gracefully when process suspension is attempted. When a process is stopped, the shell immediately executes the next command in the sequence. It suffices to place the sequence of commands between parentheses to force it into a subshell, which may be stopped as a unit.

Commands inside of  $\$(...)$  command substitution are not parsed until substitution is attempted. This will delay error reporting until some time after the command is entered. For example, unmatched parentheses, even inside shell comments, will result in error messages while the construct is being read.

Array variables may not (yet) be exported.

**NAME**

**batteryalert** – monitor battery capacity and alert to terminals on low charge

**SYNOPSIS**

root user:

**edrc/bin/batteryalert [ -h | -s | -i | -a ]**

non-root user:

**edrc/bin/batteryalert -h | -i | -a**

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

the **batteryalert** command runs in the background, monitors the power supply battery capacity and alerts (using **wall**) to all terminals on low charge.

**OPTIONS**

start **batteryalert** background process.

**-h** usage.

**-s** stop **batteryalert** background process.

**-i** print battery and **batteryalert** information.

**-a** check if battery capacity id below threshold and discharging. If so the exit status is **8**.

**ENVIRONMENT**

**\$BATTERYALERT\_THRESHOLD**

battery charge threshold in %.

The default threshold is **20** percent.

**\$BATTERYALERT\_INTERVAL**

battery charge check interval in seconds.

The default interval is **120** seconds.

**EXIT STATUS**

- 0** no error.
- 1** no battery information found.
- 2** operating system not supported.
- 4** usage printed.
- 6** cannot stop **batteryalert**.
- 7** **batteryalert** not running.
- 8** **batteryalert** is in ALERT state (when using **-a**).

**FILES**

- /sys/class/power\_supply/BAT\*/capacity**  
current battery charge level.
- /sys/class/power\_supply/BAT\*/status**  
status (**Charging/Discharging**) of the battery.
- edrc/var/settings/batteryalert.threshold**  
holds the battery alert threshold during **batteryalert** run if not default.

**EXAMPLES**

-

**SEE ALSO**

**edrcintro(1), edrcinit(1m), acpi(1), wall(1)**

**NOTES**

A service handler (**WA2Ledrc.batteryalert**) for **edrcinit(1m)** is provided.

**BUGS**

The first battery found is used for alerting and information display.

**AUTHOR**

batteryalert was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.



**NAME**

**bc** - An arbitrary precision calculator language

**SYNTAX**

**bc** [ **-hlwsqv** ] [long-options] [ *file* ... ]

**DESCRIPTION**

**bc** is a language that supports arbitrary precision numbers with interactive execution of statements. There are some similarities in the syntax to the C programming language. A standard math library is available by command line option. If requested, the math library is defined before processing any files. **bc** starts by processing code from all the files listed on the command line in the order listed. After all files have been processed, **bc** reads from the standard input. All code is executed as it is read. (If a file contains a command to halt the processor, **bc** will never read from the standard input.)

This version of **bc** contains several extensions beyond traditional **bc** implementations and the POSIX draft standard. Command line options can cause these extensions to print a warning or to be rejected. This document describes the language accepted by this processor. Extensions will be identified as such.

**OPTIONS**

- h, --help  
Print the usage and exit.
- i, --interactive  
Force interactive mode.
- l, --mathlib  
Define the standard math library.
- w, --warn  
Give warnings for extensions to POSIX **bc**.
- s, --standard  
Process exactly the POSIX **bc** language.
- q, --quiet  
Do not print the normal GNU **bc** welcome.
- v, --version  
Print the version number and copyright and quit.

**NUMBERS**

The most basic element in **bc** is the number. Numbers are arbitrary precision numbers. This precision is both in the integer part and the fractional part. All numbers are represented internally in decimal and all computation is done in decimal. (This version truncates results from divide and multiply operations.) There are two attributes of numbers, the length and the scale. The length is the total number of decimal digits used by **bc** to represent a number and the scale is the total number of decimal digits after the decimal point. For example:

.000001 has a length of 6 and scale of 6.  
1935.000 has a length of 7 and a scale of 3.

**VARIABLES**

Numbers are stored in two types of variables, simple variables and arrays. Both simple variables and array variables are named. Names begin with a letter followed by any number of letters, digits and underscores. All letters must be lower case. (Full alpha-numeric names are an extension. In POSIX **bc** all names are a single lower case letter.) The type of variable is clear by the context because all array variable names will be followed by brackets ([]).

There are four special variables, **scale**, **ibase**, **obase**, and **last**. **scale** defines how some operations use digits after the decimal point. The default value of **scale** is 0. **ibase** and **obase** define the conversion base for input and output numbers. The default for both input and output is base 10. **last** (an extension) is a variable that has the value of the last printed number. These will be discussed in further detail where appropriate. All of these variables may have values assigned to them as well as used in expressions.

## COMMENTS

Comments in **bc** start with the characters **/\*** and end with the characters **\*/**. Comments may start anywhere and appear as a single space in the input. (This causes comments to delimit other input items. For example, a comment can not be found in the middle of a variable name.) Comments include any newlines (end of line) between the start and the end of the comment.

To support the use of scripts for **bc**, a single line comment has been added as an extension. A single line comment starts at a **#** character and continues to the next end of the line. The end of line character is not part of the comment and is processed normally.

## EXPRESSIONS

The numbers are manipulated by expressions and statements. Since the language was designed to be interactive, statements and expressions are executed as soon as possible. There is no "main" program. Instead, code is executed as it is encountered. (Functions, discussed in detail later, are defined when encountered.)

A simple expression is just a constant. **bc** converts constants into internal decimal numbers using the current input base, specified by the variable **ibase**. (There is an exception in functions.) The legal values of **ibase** are 2 through 36. (Bases greater than 16 are an extension.) Assigning a value outside this range to **ibase** will result in a value of 2 or 36. Input numbers may contain the characters 0–9 and A–Z. (Note: They must be capitals. Lower case letters are variable names.) Single digit numbers always have the value of the digit regardless of the value of **ibase**. (i.e. A = 10.) For multi-digit numbers, **bc** changes all input digits greater or equal to **ibase** to the value of **ibase-1**. This makes the number **ZZZ** always be the largest 3 digit number of the input base.

Full expressions are similar to many other high level languages. Since there is only one kind of number, there are no rules for mixing types. Instead, there are rules on the scale of expressions. Every expression has a scale. This is derived from the scale of original numbers, the operation performed and in many cases, the value of the variable **scale**. Legal values of the variable **scale** are 0 to the maximum number representable by a C integer.

In the following descriptions of legal expressions, "expr" refers to a complete expression and "var" refers to a simple or an array variable. A simple variable is just a

*name*

and an array variable is specified as

*name[expr]*

Unless specifically mentioned the scale of the result is the maximum scale of the expressions involved.

- expr    The result is the negation of the expression.

++ var    The variable is incremented by one and the new value is the result of the expression.

-- var    The variable is decremented by one and the new value is the result of the expression.

var ++

          The result of the expression is the value of the variable and then the variable is incremented by one.

var --    The result of the expression is the value of the variable and then the variable is decremented by one.

expr + expr

          The result of the expression is the sum of the two expressions.

expr - expr

          The result of the expression is the difference of the two expressions.

expr \* expr

          The result of the expression is the product of the two expressions.

expr / expr

          The result of the expression is the quotient of the two expressions. The scale of the result is the value of the variable **scale**.

`expr % expr`

The result of the expression is the "remainder" and it is computed in the following way. To compute `a%b`, first `a/b` is computed to **scale** digits. That result is used to compute `a-(a/b)*b` to the scale of the maximum of **scale**+`scale(b)` and `scale(a)`. If **scale** is set to zero and both expressions are integers this expression is the integer remainder function.

`expr ^ expr`

The result of the expression is the value of the first raised to the second. The second expression must be an integer. (If the second expression is not an integer, a warning is generated and the expression is truncated to get an integer value.) The scale of the result is **scale** if the exponent is negative. If the exponent is positive the scale of the result is the minimum of the scale of the first expression times the value of the exponent and the maximum of **scale** and the scale of the first expression. (e.g. `scale(a^b) = min(scale(a)*b, max( scale, scale(a)))`.) It should be noted that `expr^0` will always return the value of 1.

( `expr` ) This alters the standard precedence to force the evaluation of the expression.

`var = expr`

The variable is assigned the value of the expression.

`var <op>= expr`

This is equivalent to "`var = var <op> expr`" with the exception that the "`var`" part is evaluated only once. This can make a difference if "`var`" is an array.

Relational expressions are a special kind of expression that always evaluate to 0 or 1, 0 if the relation is false and 1 if the relation is true. These may appear in any legal expression. (POSIX `bc` requires that relational expressions are used only in `if`, `while`, and `for` statements and that only one relational test may be done in them.) The relational operators are

`expr1 < expr2`

The result is 1 if `expr1` is strictly less than `expr2`.

`expr1 <= expr2`

The result is 1 if `expr1` is less than or equal to `expr2`.

`expr1 > expr2`

The result is 1 if `expr1` is strictly greater than `expr2`.

`expr1 >= expr2`

The result is 1 if `expr1` is greater than or equal to `expr2`.

`expr1 == expr2`

The result is 1 if `expr1` is equal to `expr2`.

`expr1 != expr2`

The result is 1 if `expr1` is not equal to `expr2`.

Boolean operations are also legal. (POSIX `bc` does NOT have boolean operations). The result of all boolean operations are 0 and 1 (for false and true) as in relational expressions. The boolean operators are:

`!expr` The result is 1 if `expr` is 0.

`expr && expr`

The result is 1 if both expressions are non-zero.

`expr || expr`

The result is 1 if either expression is non-zero.

The expression precedence is as follows: (lowest to highest)

- `||` operator, left associative
- `&&` operator, left associative
- `!` operator, nonassociative
- Relational operators, left associative
- Assignment operator, right associative

+ and - operators, left associative  
 \*, / and % operators, left associative  
 ^ operator, right associative  
 unary - operator, nonassociative  
 ++ and -- operators, nonassociative

This precedence was chosen so that POSIX compliant **bc** programs will run correctly. This will cause the use of the relational and logical operators to have some unusual behavior when used with assignment expressions. Consider the expression:

```
a = 3 < 5
```

Most C programmers would assume this would assign the result of "3 < 5" (the value 1) to the variable "a". What this does in **bc** is assign the value 3 to the variable "a" and then compare 3 to 5. It is best to use parenthesis when using relational and logical operators with the assignment operators.

There are a few more special expressions that are provided in **bc**. These have to do with user defined functions and standard functions. They all appear as "*name(parameters)*". See the section on functions for user defined functions. The standard functions are:

length ( expression )

The value of the length function is the number of significant digits in the expression.

read ( ) The read function (an extension) will read a number from the standard input, regardless of where the function occurs. Beware, this can cause problems with the mixing of data and program in the standard input. The best use for this function is in a previously written program that needs input from the user, but never allows program code to be input from the user. The value of the read function is the number read from the standard input using the current value of the variable **ibase** for the conversion base.

scale ( expression )

The value of the scale function is the number of digits after the decimal point in the expression.

sqrt ( expression )

The value of the sqrt function is the square root of the expression. If the expression is negative, a run time error is generated.

## STATEMENTS

Statements (as in most algebraic languages) provide the sequencing of expression evaluation. In **bc** statements are executed "as soon as possible." Execution happens when a newline is encountered and there is one or more complete statements. Due to this immediate execution, newlines are very important in **bc**. In fact, both a semicolon and a newline are used as statement separators. An improperly placed newline will cause a syntax error. Because newlines are statement separators, it is possible to hide a newline by using the backslash character. The sequence "\<nl>", where <nl> is the newline appears to **bc** as whitespace instead of a newline. A statement list is a series of statements separated by semicolons and newlines. The following is a list of **bc** statements and what they do: (Things enclosed in brackets []) are optional parts of the statement.)

expression

This statement does one of two things. If the expression starts with "<variable> <assignment> ...", it is considered to be an assignment statement. If the expression is not an assignment statement, the expression is evaluated and printed to the output. After the number is printed, a newline is printed. For example, "a=1" is an assignment statement and "(a=1)" is an expression that has an embedded assignment. All numbers that are printed are printed in the base specified by the variable **obase**. The legal values for **obase** are 2 through BC\_BASE\_MAX. (See the section LIMITS.) For bases 2 through 16, the usual method of writing numbers is used. For bases greater than 16, **bc** uses a multi-character digit method of printing the numbers where each higher base digit is printed as a base 10 number. The multi-character digits are separated by spaces. Each digit contains the number of characters required to represent the base ten value of "obase-1". Since numbers are of arbitrary precision, some numbers may not be printable on a single output line. These long numbers will be split across lines using the "\" as the last character on a line. The maximum

number of characters printed per line is 70. Due to the interactive nature of **bc**, printing a number causes the side effect of assigning the printed value to the special variable **last**. This allows the user to recover the last value printed without having to retype the expression that printed the number. Assigning to **last** is legal and will overwrite the last printed value with the assigned value. The newly assigned value will remain until the next number is printed or another value is assigned to **last**. (Some installations may allow the use of a single period (.) which is not part of a number as a short hand notation for for **last**.)

**string** The string is printed to the output. Strings start with a double quote character and contain all characters until the next double quote character. All characters are take literally, including any newline. No newline character is printed after the string.

#### **print list**

The print statement (an extension) provides another method of output. The "list" is a list of strings and expressions separated by commas. Each string or expression is printed in the order of the list. No terminating newline is printed. Expressions are evaluated and their value is printed and assigned to the variable **last**. Strings in the print statement are printed to the output and may contain special characters. Special characters start with the backslash character (\). The special characters recognized by **bc** are "a" (alert or bell), "b" (backspace), "f" (form feed), "n" (newline), "r" (carriage return), "q" (double quote), "t" (tab), and "\" (backslash). Any other character following the backslash will be ignored.

**{ statement\_list }**

This is the compound statement. It allows multiple statements to be grouped together for execution.

**if ( expression ) statement1 [else statement2]**

The if statement evaluates the expression and executes statement1 or statement2 depending on the value of the expression. If the expression is non-zero, statement1 is executed. If statement2 is present and the value of the expression is 0, then statement2 is executed. (The else clause is an extension.)

**while ( expression ) statement**

The while statement will execute the statement while the expression is non-zero. It evaluates the expression before each execution of the statement. Termination of the loop is caused by a zero expression value or the execution of a break statement.

**for ( [expression1] ; [expression2] ; [expression3] ) statement**

The for statement controls repeated execution of the statement. Expression1 is evaluated before the loop. Expression2 is evaluated before each execution of the statement. If it is non-zero, the statement is evaluated. If it is zero, the loop is terminated. After each execution of the statement, expression3 is evaluated before the reevaluation of expression2. If expression1 or expression3 are missing, nothing is evaluated at the point they would be evaluated. If expression2 is missing, it is the same as substituting the value 1 for expression2. (The optional expressions are an extension. POSIX **bc** requires all three expressions.) The following is equivalent code for the for statement:

```
expression1;
while (expression2) {
 statement;
 expression3;
}
```

**break** This statement causes a forced exit of the most recent enclosing while statement or for statement.

#### **continue**

The continue statement (an extension) causes the most recent enclosing for statement to start the next iteration.

#### **halt**

The halt statement (an extension) is an executed statement that causes the **bc** processor to quit only when it is executed. For example, "if (0 == 1) halt" will not cause **bc** to terminate because the halt is not executed.

**return** Return the value 0 from a function. (See the section on functions.)

**return** ( expression )

Return the value of the expression from a function. (See the section on functions.) As an extension, the parenthesis are not required.

## PSEUDO STATEMENTS

These statements are not statements in the traditional sense. They are not executed statements. Their function is performed at "compile" time.

**limits** Print the local limits enforced by the local version of **bc**. This is an extension.

**quit** When the quit statement is read, the **bc** processor is terminated, regardless of where the quit statement is found. For example, "if (0 == 1) quit" will cause **bc** to terminate.

**warranty**

Print a longer warranty notice. This is an extension.

## FUNCTIONS

Functions provide a method of defining a computation that can be executed later. Functions in **bc** always compute a value and return it to the caller. Function definitions are "dynamic" in the sense that a function is undefined until a definition is encountered in the input. That definition is then used until another definition for the same name is encountered. The new definition then replaces the older definition. A function is defined as follows:

```
define name (parameters) { newline
 auto_list statement_list }
```

A function call is just an expression of the form "*name(parameters)*".

Parameters are numbers or arrays (an extension). In the function definition, zero or more parameters are defined by listing their names separated by commas. All parameters are call by value parameters. Arrays are specified in the parameter definition by the notation "*name*[]". In the function call, actual parameters are full expressions for number parameters. The same notation is used for passing arrays as for defining array parameters. The named array is passed by value to the function. Since function definitions are dynamic, parameter numbers and types are checked when a function is called. Any mismatch in number or types of parameters will cause a runtime error. A runtime error will also occur for the call to an undefined function.

The *auto\_list* is an optional list of variables that are for "local" use. The syntax of the auto list (if present) is "**auto** *name*, ... ;". (The semicolon is optional.) Each *name* is the name of an auto variable. Arrays may be specified by using the same notation as used in parameters. These variables have their values pushed onto a stack at the start of the function. The variables are then initialized to zero and used throughout the execution of the function. At function exit, these variables are popped so that the original value (at the time of the function call) of these variables are restored. The parameters are really auto variables that are initialized to a value provided in the function call. Auto variables are different than traditional local variables because if function A calls function B, B may access function A's auto variables by just using the same name, unless function B has called them auto variables. Due to the fact that auto variables and parameters are pushed onto a stack, **bc** supports recursive functions.

The function body is a list of **bc** statements. Again, statements are separated by semicolons or newlines. Return statements cause the termination of a function and the return of a value. There are two versions of the return statement. The first form, "**return**", returns the value 0 to the calling expression. The second form, "**return** ( *expression* )", computes the value of the expression and returns that value to the calling expression. There is an implied "**return** (0)" at the end of every function. This allows a function to terminate and return 0 without an explicit return statement.

Functions also change the usage of the variable **ibase**. All constants in the function body will be converted using the value of **ibase** at the time of the function call. Changes of **ibase** will be ignored during the execution of the function except for the standard function **read**, which will always use the current value of **ibase** for conversion of numbers.

Several extensions have been added to functions. First, the format of the definition has been slightly

relaxed. The standard requires the opening brace be on the same line as the **define** keyword and all other parts must be on following lines. This version of **bc** will allow any number of newlines before and after the opening brace of the function. For example, the following definitions are legal.

```
define d (n) { return (2*n); }
define d (n)
{ return (2*n); }
```

Functions may be defined as **void**. A void function returns no value and thus may not be used in any place that needs a value. A void function does not produce any output when called by itself on an input line. The key word **void** is placed between the key word **define** and the function name. For example, consider the following session.

```
define py (y) { print "---->", y, "<---", "\n"; }
define void px (x) { print "---->", x, "<---", "\n"; }
py(1)
---->1<---
0
px(1)
---->1<---
```

Since **py** is not a void function, the call of **py(1)** prints the desired output and then prints a second line that is the value of the function. Since the value of a function that is not given an explicit return statement is zero, the zero is printed. For **px(1)**, no zero is printed because the function is a void function.

Also, call by variable for arrays was added. To declare a call by variable array, the declaration of the array parameter in the function definition looks like **"\*name[]"**. The call to the function remains the same as call by value arrays.

## MATH LIBRARY

If **bc** is invoked with the **-l** option, a math library is preloaded and the default scale is set to 20. The math functions will calculate their results to the scale set at the time of their call. The math library defines the following functions:

- s (x)** The sine of x, x is in radians.
- c (x)** The cosine of x, x is in radians.
- a (x)** The arctangent of x, arctangent returns radians.
- l (x)** The natural logarithm of x.
- e (x)** The exponential function of raising e to the value x.
- j (n,x)** The Bessel function of integer order n of x.

## EXAMPLES

In **/bin/sh**, the following will assign the value of "pi" to the shell variable **pi**.

```
pi=$(echo "scale=10; 4*a(1)" | bc -l)
```

The following is the definition of the exponential function used in the math library. This function is written in POSIX **bc**.

```
scale = 20

/* Uses the fact that e^x = (e^(x/2))^2
 When x is small enough, we use the series:
 e^x = 1 + x + x^2/2! + x^3/3! + ...
*/

define e(x) {
 auto a, d, e, f, i, m, v, z

 /* Check the sign of x. */
 if (x<0) {
```

```

 m = 1
 x = -x
}

/* Precondition x. */
z = scale;
scale = 4 + z + .44*x;
while (x > 1) {
 f += 1;
 x /= 2;
}

/* Initialize the variables. */
v = 1+x
a = x
d = 1

for (i=2; 1; i++) {
 e = (a *= x) / (d *= i)
 if (e == 0) {
 if (f>0) while (f--) v = v*v;
 scale = z
 if (m) return (1/v);
 return (v/1);
 }
 v += e
}
}

```

The following is code that uses the extended features of **bc** to implement a simple program for calculating checkbook balances. This program is best kept in a file so that it can be used many times without having to retype it at every use.

```

scale=2
print "\nCheck book program!\n"
print " Remember, deposits are negative transactions.\n"
print " Exit by a 0 transaction.\n\n"

print "Initial balance? "; bal = read()
bal /= 1
print "\n"
while (1) {
 "current balance = "; bal
 "transaction? "; trans = read()
 if (trans == 0) break;
 bal -= trans
 bal /= 1
}
quit

```

The following is the definition of the recursive factorial function.

```

define f (x) {
 if (x <= 1) return (1);
 return (f(x-1) * x);
}

```



## READLINE AND LIBEDIT OPTIONS

GNU **bc** can be compiled (via a configure option) to use the GNU **readline** input editor library or the BSD **libedit** library. This allows the user to do editing of lines before sending them to **bc**. It also allows for a history of previous lines typed. When this option is selected, **bc** has one more special variable. This special variable, **history** is the number of lines of history retained. For **readline**, a value of `-1` means that an unlimited number of history lines are retained. Setting the value of **history** to a positive number restricts the number of history lines to the number given. The value of `0` disables the history feature. The default value is `100`. For more information, read the user manuals for the GNU **readline**, **history** and BSD **libedit** libraries. One can not enable both **readline** and **libedit** at the same time.

## DIFFERENCES

This version of **bc** was implemented from the POSIX P1003.2/D11 draft and contains several differences and extensions relative to the draft and traditional implementations. It is not implemented in the traditional way using *dc(1)*. This version is a single process which parses and runs a byte code translation of the program. There is an "undocumented" option (`-c`) that causes the program to output the byte code to the standard output instead of running it. It was mainly used for debugging the parser and preparing the math library.

A major source of differences is extensions, where a feature is extended to add more functionality and additions, where new features are added. The following is the list of differences and extensions.

### LANG environment

This version does not conform to the POSIX standard in the processing of the LANG environment variable and all environment variables starting with `LC_`.

**names** Traditional and POSIX **bc** have single letter names for functions, variables and arrays. They have been extended to be multi-character names that start with a letter and may contain letters, numbers and the underscore character.

**Strings** Strings are not allowed to contain NUL characters. POSIX says all characters must be included in strings.

**last** POSIX **bc** does not have a **last** variable. Some implementations of **bc** use the period (`.`) in a similar way.

### comparisons

POSIX **bc** allows comparisons only in the if statement, the while statement, and the second expression of the for statement. Also, only one relational operation is allowed in each of those statements.

### if statement, else clause

POSIX **bc** does not have an else clause.

### for statement

POSIX **bc** requires all expressions to be present in the for statement.

### &&, ||, !

POSIX **bc** does not have the logical operators.

### read function

POSIX **bc** does not have a read function.

### print statement

POSIX **bc** does not have a print statement.

### continue statement

POSIX **bc** does not have a continue statement.

### return statement

POSIX **bc** requires parentheses around the return expression.

**array parameters**

POSIX **bc** does not (currently) support array parameters in full. The POSIX grammar allows for arrays in function definitions, but does not provide a method to specify an array as an actual parameter. (This is most likely an oversight in the grammar.) Traditional implementations of **bc** have only call by value array parameters.

**function format**

POSIX **bc** requires the opening brace on the same line as the **define** key word and the **auto** statement on the next line.

**==+, ==-, ==\*, ==/, ==%, ==^**

POSIX **bc** does not require these "old style" assignment operators to be defined. This version may allow these "old style" assignments. Use the limits statement to see if the installed version supports them. If it does support the "old style" assignment operators, the statement "a ==- 1" will decrement **a** by 1 instead of setting **a** to the value -1.

**spaces in numbers**

Other implementations of **bc** allow spaces in numbers. For example, "x=1 3" would assign the value 13 to the variable **x**. The same statement would cause a syntax error in this version of **bc**.

**errors and execution**

This implementation varies from other implementations in terms of what code will be executed when syntax and other errors are found in the program. If a syntax error is found in a function definition, error recovery tries to find the beginning of a statement and continue to parse the function. Once a syntax error is found in the function, the function will not be callable and becomes undefined. Syntax errors in the interactive execution code will invalidate the current execution block. The execution block is terminated by an end of line that appears after a complete sequence of statements. For example,

```
a = 1
b = 2
```

has two execution blocks and

```
{ a = 1
 b = 2 }
```

has one execution block. Any runtime error will terminate the execution of the current execution block. A runtime warning will not terminate the current execution block.

**Interrupts**

During an interactive session, the SIGINT signal (usually generated by the control-C character from the terminal) will cause execution of the current execution block to be interrupted. It will display a "runtime" error indicating which function was interrupted. After all runtime structures have been cleaned up, a message will be printed to notify the user that **bc** is ready for more input. All previously defined functions remain defined and the value of all non-auto variables are the value at the point of interruption. All auto variables and function parameters are removed during the clean up process. During a non-interactive session, the SIGINT signal will terminate the entire run of **bc**.

**LIMITS**

The following are the limits currently in place for this **bc** processor. Some of them may have been changed by an installation. Use the limits statement to see the actual values.

**BC\_BASE\_MAX**

The maximum output base is currently set at 999. The maximum input base is 16.

**BC\_DIM\_MAX**

This is currently an arbitrary limit of 65535 as distributed. Your installation may be different.

**BC\_SCALE\_MAX**

The number of digits after the decimal point is limited to INT\_MAX digits. Also, the number of digits before the decimal point is limited to INT\_MAX digits.

**BC\_STRING\_MAX**

The limit on the number of characters in a string is INT\_MAX characters.

**exponent**

The value of the exponent in the raise operation (\*\*) is limited to LONG\_MAX.

**variable names**

The current limit on the number of unique names is 32767 for each of simple variables, arrays and functions.

**ENVIRONMENT VARIABLES**

The following environment variables are processed by **bc**:

**POSIXLY\_CORRECT**

This is the same as the **-s** option.

**BC\_ENV\_ARGS**

This is another mechanism to get arguments to **bc**. The format is the same as the command line arguments. These arguments are processed first, so any files listed in the environment arguments are processed before any command line argument files. This allows the user to set up "standard" options and files to be processed at every invocation of **bc**. The files in the environment variables would typically contain function definitions for functions the user wants defined every time **bc** is run.

**BC\_LINE\_LENGTH**

This should be an integer specifying the number of characters in an output line for numbers. This includes the backslash and newline characters for long numbers. As an extension, the value of zero disables the multi-line feature. Any other value of this variable that is less than 3 sets the line length to 70.

**DIAGNOSTICS**

If any file on the command line can not be opened, **bc** will report that the file is unavailable and terminate. Also, there are compile and run time diagnostics that should be self-explanatory.

**BUGS**

Error recovery is not very good yet.

Email bug reports to **bug-bc@gnu.org**. Be sure to include the word "bc" somewhere in the "Subject:" field.

**AUTHOR**

Philip A. Nelson  
philnelson@acm.org

**ACKNOWLEDGEMENTS**

The author would like to thank Steve Sommars (Steve.Sommars@att.com) for his extensive help in testing the implementation. Many great suggestions were given. This is a much better product due to his involvement.

**NAME**

bget – basic HTTP get tool

**DESCRIPTION**

Basic tool to make HTTP GET requests and monitor the results. Unlike LWP GET, it does not require special Perl modules, and by virtue of being cruder makes HTTP headers easier to spy on.

Only URLs of the forms

```
http://hostname/[localpart]
http://hostname:port/[localpart]
```

are supported.

Options:

- `-a --autoname`

Save output automatically based on URI. Will not warn if the file already exists. This overrides the `-o` (`--out`) option. The preferred output name is everything after the last `/` in the URL, or `'dir-default'` if the URL ends with a `/`.

- `-B --no-body`

Don't print the body of the response.

- `-b --browser NAME`

What browser to emulate. Use `--emulations` to list available browser headers.

- `-c --cookie VALUE`

Set the cookie header with VALUE.

- `-d --dontdechunk`

As of version 1.2, when the response headers indicate a Transfer-Encoding of 'chunked', bget will rename the header (prefixing it with 'Xbget-') and unchunk the response. This is desirable so that chunked responses from HTTP/1.1 servers look right. In some cases it may be desirable to see raw output from the server however, so this behavior can be turned off.

- `-e --head`

Make a HEAD request instead of a GET. Note that this does not imply `-h` (`--heads`) to print the headers, nor `-B` (`--no-body`) to suppress printing any body content. (Some servers, eg `www.yahoo.com`, treat HEAD like a GET.)

- `-F --file FILE`

Read URLs from FILE (one per line) instead of from command line. Use filename `-` for standard input.

If there are two URLs on a line, the first one is used as the referer URL. The referer will remain unused until the next line with two URLs.

If there is an additional field after the URL, that will be used as an `-o` (`--out`) output file until the next line with an output file. An output file should not begin with `"http:/"` or `"https:/"`.

Fields on each line of the URL file are whitespace separated.

- `-f --follow`

Follow redirects. If printing headers, the redirecting headers and the destination headers will be printed. (No loop detection is attempted.) If printing bodies and not saving via `autoname`, the redirecting body and the destination body will be printed. If saving via `autoname`, a new file will be opened for each request made. Some redirects (eg loops) may cause the `autoname` to pick the same filename as a previous request, which will cause the earlier file to be clobbered.

- `-H --host HOST[:P]`  
Connect to HOST for request (useful for testing virtual hosts before a DNS change or use with `-l` for proxies).
- `-h --heads`  
Print the response headers.
- `-L --language LANG`  
Use LANG for Accept-Language: header. See `--languages` for a small list.
- `-l --long`  
Use long address on GET line (using the full `http://...` format, a MUST for HTTP/1.1 server compliance but handy with `-H` for proxies).
- `-o --out FILE`  
Write output to FILE. Unlike `-a` (`--autoname`) this will not use a different file for each request. The autoname option has precedence over this option. Filenames in a `-F` (`--file`) URL file will also override this.
- `-p --post STRING`  
Use STRING as a post form contents (forms of type `application/x-www-form-urlencoded` only).
- `-P --filepost FILE`  
Use contents of FILE as a post form contents. If the first line is of the form “Content-Type: foo/bar” it will be used to set the Content-Type: header. More than just the MIME type is allowed, but it must be all on one line. Typical POST content types are  

```
application/x-www-form-urlencoded
 Encoded like a typical CGI URL.
multipart/form-data
 Each form element is in a separate MIME part; needed for
 file uploads. This type requires a boundary parameter on
 the Content-Type: header.
```

There is a similar allowance for setting the Transfer-Encoding: header. This must be on the second line if Content-Type: is set, and on the first line if not. When Transfer-Encoding: contains the string ‘chunked’, Content-Length: will not be set for a post. Note that apache 1.3.x (at least) does not allow chunked POST requests.

You may find the tool `mkpost` (available in the scripts section of CPAN) to be helpful in creating CGI interface files for this option. For other content types, like “text/xml” for XML-RPC interface requests, other tools will be needed.
- `-R --refer VALUE`  
Set the (initial) referer header with VALUE.
- `-r --request`  
Print the request headers.
- `-s --status CODE`  
After fetching a page — including following redirects and printing bits of the response as controlled by other options — if the HTTP status code is not exactly the one given, bget will exit (returning code 3 to the shell). Useful for looping until one hits a 404 or the like.

- `-t --time N`  
Use Benchmark module to time making the command line request(s) N times.
- `-C --count N`  
Just like `-t/--time`, but optimizations apply: if neither heads nor bodies are requested, nothing will be fetched. If body is not requested only heads will be fetched.
- `-u --user USER:PW`  
Basic authentication in the form {username}:{password}.
- `-w --wait N`  
Wait N seconds between fetching each URL.
- `-w --wait A,D`  
Waits a random number of seconds, average A standard deviation D, between fetching each URL. Requires the Math::Random module. Useful for being subtle when fetching a lot of pages, along with emulating a browser and using per-page referer headers via the `-F (--file)` method.
- `--help`  
Show a help message and exit.
- `--version`  
Print version and exit.
- `--emulations`  
Print list of available browser emulations.
- `--languages`  
Print a sample of language codes.

#### Note

If `-H (--host)` is used with multiple URLs, all connections are made to the specified HOST (and port) even if different hosts are used in the URLs. This can be used to fetch files through a HTTP proxy if `-l (--long)` is also used.

With `-L (--langauge)` the Accept-Language: header will not be added if the browser has not been observed to use it.

## EMULATIONS

The following browsers are recognized for header emulation. This might not be the definitive list. Check `--emulations` for that. Some have comments to help identify them.

- Amaya-8.1  
Amaya is the W3C's combination browser page editor.
- links-0.84  
Text mode browser for Unix. <http://artax.karlin.mff.cuni.cz/~mikulas/links> Version 0.84 does not do cookies or referer headers, so we might misemulate it that way.
- elinks-0.5pre4-linux  
Forked from links, this is another text mode browser. Quirks include giving a bunch away about the system, including window size, in the User-Agent: and including a 'Referer' header in URLs entered by hand. The User-Agent for this is from a Redhat 7.1 x86 system in an 80x24 window. <http://elinks.or.cz/>
- w3c-5.2.8

Command line web tool that uses libwww. <<http://www.w3.org/ComLine/>>

- w3m-beta99

Text mode browser for Unix. <<http://ei5nazha.yz.yamagata-u.ac.jp/~aito/w3m/>>

- Dillo-0.8.4

Dillo is Linux browser, under current development, that focuses on speed, small size, and protocol correctness. It can do cookies, but it defaults to not accepting them. It does not do Referer: headers, but bget may misemulate it on that point. <<http://www.dillo.org/>>

- Linux-Mosaic-2.6

The browser that started the rush, compiled for Linux. This is an archaic browser. It doesn't do Host: headers or Cookies:. bget can misemulate the Cookies: part, but won't do the Host: header. Many modern sites require this for proper operation, so expect problems. The headers this thing spits out are longer even than the Lynx ones.

- Qweb-1.3

Qweb was an early X11 style-sheet capable browser. Too bad it didn't do javascript (needed for some stylesheets) or even Host: headers. bget will misemulate this if you use Cookies, but won't supply a Host: header.

- X11-Chimera-1.70

The name 'Chimera' has been used by two different browsers. This is the X11 Chimera developed at the University of Las Vegas, not the Mac Mozilla derivative Chimera. In authentic use this browser does not have cookies or use Referer: headers.

- ApacheBench-1.3

ab, the benchmark tool that comes with the Apache httpd package.

- Opera-3.60

An old version of a popular alternative browser for Windows.

- Windows-Opera-7beta

More modern (2003) version of Opera.

- Linux-Opera-6.11

As of Opera 6.x there is a linux version.

- lwp-request-1.38

Lib WWW Perl module (these are the default headers).

- wget-1.6

Command-line bulk page downloading tool for Unix.

- NetBSD-curl-7.10.4-HTTP1.1

Command-line page upload/download tool for Unix. Prefers HTTP/1.1 but can do HTTP/1.0 upon request. Can do PUTs and DELETes and other obscure things, too.

- NetBSD-curl-7.10.4-HTTP1.0

Curl in HTTP/1.0 mode.

- iCab-pre1.7

Popular alternative browser for Macs.

- junkbuster-2

Once popular ad- and cookie-filtering proxy. Junkbuster does a bunch of header editing from the

actual browser headers, and thus the headers out of it can vary considerably from this. It looks like Accept-\* headers are not edited, allowing identification of the underlying browser sometimes. The Accept-\* headers here come from a Netscape 4.7. By default, Junkbuster masquerades as Netscape 3.01 (GOLD) for Mac PPC.

- **Lynx-2.8.1**  
Popular text mode browser, predominately unix.
- **Linux-Mozilla-1.0.0**  
Mozilla is the open source version of Netscape 7. It exists for many platforms.
- **Linux-Phoenix-0.6-beta**  
Phoenix (formerly Firebird) is Mozilla with a different user-interface library. There are unix, windows and mac variants.
- **Konqueror-2.1.1**  
Konqueror is a mostly-Linux browser based on KDE.
- **OpenOffice-1.0.0**  
OpenOffice is a StarOffice relation, intended to be a free Unix "Office" compatible software bundle. It includes an HTML editor that can download pages to edit, but as such it does things like issue PROPFIND requests that are not emulated here.
- **WindowsNT-Explorer-5.0-as-4.0**  
Explorer 5.0 can be installed with a compatibility mode that emulates (or claims to emulate) Explorer 4.0.
- **Windows98-Explorer-5.5**
- **WindowsNT-ActiveDesktop**  
This is on a system with IE5.5 installed, but this identifies itself as IE4.01. This one is hard to do right, since in my tests I saw two requests for the test file. The first came with this UA, the second had this instead:  
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; MSIECrawler; Windows NT)  
The crawler version had an 'Accept-Language: us-en' as well as a different order to the headers (Accept: User-Agent:, Accept-Language: Accept-Encoding, Host:).
- **WindowsNT-Netscape6**
- **WindowsNT-Explorer-5.5**
- **Windows98-Explorer-4.0**
- **WindowsNT-Explorer-5.0**  
Normal mode Windows NT IE 5.0.
- **WindowsNT-ExplorerOffline-5.0**  
IE can optionally crawl pages to cache them for offline browsing. This is Windows NT IE 5.01 in crawl mode.
- **WindowsNT-Netscape-4.6**
- **MacPPC-Explorer-4.0**  
Mac PPC is System 7, 8 or 9 on PowerPC computers.
- **MacPPC-Netscape-4.0**
- **MacPPC-Netscape-4.6**



- MacOSX-Safari-1.2.4  
Safari is a Mozilla derivative that ships with OS X.
- MacOSX-Explorer-5.2  
Internet Explorer for OS X. (Comes with OS X?)
- Linux-Netscape-3.0
- Linux-Netscape-4.51

## LANGUAGES

In HTTP standard languages use the ISO 639 two letter code, but can have an optional two letter country code for national variants. Generic English is 'en', American English is 'en-us', Irish English is 'en-ie', Australian English is 'en-au'.

Some other lanuages:

|       |                    |
|-------|--------------------|
| af    | Afrikaans          |
| sq    | Albanian           |
| eu    | Basque             |
| bg    | Bulgarian          |
| be    | Byelorussian       |
| ca    | Catalan            |
| zh    | Chinese            |
| zh-cn | Chinese/China      |
| zh-tw | Chinese/Taiwan     |
| hr    | Croatian           |
| cs    | Czech              |
| da    | Danish             |
| nl    | Dutch              |
| nl-be | Dutch/Belgium      |
| fo    | Faeroese           |
| fi    | Finnish            |
| fr    | French             |
| fr-be | French/Belgium     |
| fr-ca | French/Canada      |
| fr-fr | French/France      |
| fr-ch | French/Switzerland |
| gl    | Galician           |
| de    | German             |
| de-at | German/Austria     |
| de-de | German/Germany     |
| de-ch | German/Switzerland |
| el    | Greek              |
| hu    | Hungarian          |
| is    | Icelandic          |
| id    | Indonesian         |
| ga    | Irish              |
| it    | Italian            |
| ja    | Japanese           |
| ko    | Korean             |
| mk    | Macedonian         |
| no    | Norwegian          |
| pl    | Polish             |
| pt    | Portuguese         |
| pt-br | Portuguese/Brazil  |
| ro    | Romanian           |
| ru    | Russian            |
| gd    | Scots Gaelic       |
| sr    | Serbian            |
| sk    | Slovak             |
| sl    | Slovenian          |
| es    | Spanish            |
| es-ar | Spanish/Argentina  |
| es-co | Spanish/Colombia   |
| ex-mx | Spanish/Mexico     |
| es-es | Spanish/Spain      |
| sv    | Swedish            |
| tr    | Turkish            |
| uk    | Ukrainian          |

This list is from the default set of languages in Netscape 4.5. IE has a different set, including more country

variations. Note that the country variations are frequently misused. A request with a language header like:

```
Accept-Language: en-us, es-mx; q=0.7, fr-ca; q=0.3
```

Would specify a first choice language of US English, second choice Mexican Spanish, third choice Canadian French. If a content-negotiating server only has generic English, generic Spanish, and generic French, then by specification it should return a “406 Not Acceptable” error, since it has no languages that match. This could be seen as a deficiency of the spec, but that’s the way it is.

## REVISION HISTORY

NEW IN VERSION 1.2

By supporting chunked transfer encodings, the author considers bget to be HTTP/1.1 compliant now. A word of warning, some emulations specify various allowed other encodings, like gzipped content. You should be prepared to deal with these outside of bget.

## SEE ALSO

mkpost — build bodies for HTTP CGI POST requests

## COPYRIGHT

Copyright 1999–2005 by Eli the Bearded / Benjamin Elijah Griffin. Released under the same license(s) as Perl.

## AUTHOR

Eli the Bearded originally wrote this to spy on headers and have a low cpu impact way to fetch files over http. It evolved from there.

## CPAN INFO

### SCRIPT CATEGORIES

Web

## README

bget – basic HTTP get tool

## PREREQUISITES

This uses the `strict`, `vars`, `Socket`, and `Carp` modules.

## COREQUISITES

This will try to use the `Benchmark` and `Math::Random` modules when run with certain options.

## OSNAMES

Should not be OS dependent. The `autoname` feature (`-a / --autoname`) assumes that `/` separates directories, however this should have minimal impact since it always tries to save in the current directory. Problems will likely only ensue if the automatically chosen name contains a directory separator for the current OS.

**NAME**

binprobe – probe compiled binaries and perl scripts

**SYNOPSIS**

**edrc/bin/binprobe** [ **-h** ]

**binprobe** [ **-l** | **-v** ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

probe (compiled) executables and perl scripts if they are able to start on the current operating system.

**OPTIONS**

- h**           usage message.
- l**           list all binaries that are probed.
- v**           verbose output of additional information.

**ENVIRONMENT**

-

**EXIT STATUS**

- 0**           no error.
- 2**           operating system not supported. See **osid**(3) if you get this error.
- 4**           usage displayed.

**FILES**

**etc/binprobe.dat**  
configuration file for **binprobe**(1m).

**EXAMPLES**

-

**SEE ALSO**

**edrcintro(1)**, **binprobe.dat(4)**, **cmdlist(1m)**, **compatibility(1)**, **glibc.version(3)**, **java\_wrapper(1)**, **lua\_wrapper(1)**, **osid(3)**, **os\_wrapper(1)**, **perl\_wrapper(1)**, **python\_wrapper(1)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

binprobe was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

binprobe.dat – binary probe configuration for binprobe

**SYNOPSIS**

edrc/etc/binprobe.dat

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

This is the binary probe definition for the **binprobe** command.

**FILEFORMAT**

The fileformat is a list of definitions that have the format

*COMMAND;TYPE;OPTIONS;EXITCODE;TESTCMD;COMMENT;*

Rows starting with a # are considered as comments. Empty lines are allowed, too.

**OPTIONS**

*COMMAND*

command to probe.

*TYPE*

command type as in **compatibility**(1).

*OPTIONS*

probe options.

*EXITCODE*

expected exit code.

*TESTCMD*

test command if it is different from *COMMAND OPTIONS*.

When the *TESTCMD* is set to **MANUAL** the *COMMAND* is not executed, but the status is set to [ **MANL** ] to indicate that a manual check is needed. This is to document GUI programs.

**COMMENT**

free text comment.

**EXAMPLES**

```
#
binprobe.dat - configuration file for binprobe
#
[00] 25.10.2020 CWa Initial Version
[01] 01.11.2020 CWa ++
#
COMMAND;TYPE;OPTIONS;EXITCODE;TESTCMD;COMMENT;
banner;BIN;-h;0;;;
bunzip2;BIN;-h;0;;;
bzcat;BIN;-h;0;;;
bzdiff;SCR;-h;1;;;
bzgrep;SCR;probe probe;1;;;
changepwd_trusted;BIN;-h;0;;HP-UX 11 only;
epub2pdf;JBIN;;;MANUAL;;
psjoin;PSCR;-h0;;;

```

**SEE ALSO**

**edrcintro(1)**, **compatibility(1)**, **binprobe(1m)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

binprobe.dat was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

**bzip2**, **bunzip2** – a block-sorting file compressor, v1.0.3  
**bzcat** – decompresses files to stdout  
**bzip2recover** – recovers data from damaged bzip2 files

**SYNOPSIS**

**bzip2** [ **-cdfkqstvwVL123456789** ] [ *filenames ...* ]  
**bunzip2** [ **-fkvsVL** ] [ *filenames ...* ]  
**bzcat** [ **-s** ] [ *filenames ...* ]  
**bzip2recover** *filename*

**DESCRIPTION**

*bzip2* compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM family of statistical compressors.

The command-line options are deliberately very similar to those of *GNU gzip*, but they are not identical.

*bzip2* expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name "original\_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

*bzip2* and *bunzip2* will by default not overwrite existing files. If you want this to happen, specify the **-f** flag.

If no file names are specified, *bzip2* compresses from standard input to standard output. In this case, *bzip2* will decline to write compressed output to a terminal, as this would be entirely incomprehensible and therefore pointless.

*bunzip2* (or *bzip2 -d*) decompresses all specified files. Files which were not created by *bzip2* will be detected and ignored, and a warning issued. *bzip2* attempts to guess the filename for the decompressed file from that of the compressed file as follows:

```
filename.bz2 becomes filename
filename.bz becomes filename
filename.tbz2 becomes filename.tar
filename.tbz becomes filename.tar
anyothername becomes anyothername.out
```

If the file does not end in one of the recognised endings, *.bz2*, *.bz*, *.tbz2* or *.tbz*, *bzip2* complains that it cannot guess the name of the original file, and uses the original name with *.out* appended.

As with compression, supplying no filenames causes decompression from standard input to standard output.

*bunzip2* will correctly decompress a file which is the concatenation of two or more compressed files. The result is the concatenation of the corresponding uncompressed files. Integrity testing (**-t**) of concatenated compressed files is also supported.

You can also compress or decompress files to the standard output by giving the **-c** flag. Multiple files may be compressed and decompressed like this. The resulting outputs are fed sequentially to stdout.



Compression of multiple files in this manner generates a stream containing multiple compressed file representations. Such a stream can be decompressed correctly only by *bzip2* version 0.9.0 or later. Earlier versions of *bzip2* will stop after decompressing the first file in the stream.

*bzcat* (or *bzip2 -dc*) decompresses all specified files to the standard output.

*bzip2* will read arguments from the environment variables *BZIP2* and *BZIP*, in that order, and will process them before any arguments read from the command line. This gives a convenient way to supply default arguments.

Compression is always performed, even if the compressed file is slightly larger than the original. Files of less than about one hundred bytes tend to get larger, since the compression mechanism has a constant overhead in the region of 50 bytes. Random data (including the output of most file compressors) is coded at about 8.05 bits per byte, giving an expansion of around 0.5%.

As a self-check for your protection, *bzip2* uses 32-bit CRCs to make sure that the decompressed version of a file is identical to the original. This guards against corruption of the compressed data, and against undetected bugs in *bzip2* (hopefully very unlikely). The chances of data corruption going undetected is microscopic, about one chance in four billion for each file processed. Be aware, though, that the check occurs upon decompression, so it can only tell you that something is wrong. It can't help you recover the original uncompressed data. You can use *bzip2recover* to try to recover data from damaged files.

Return values: 0 for a normal exit, 1 for environmental problems (file not found, invalid flags, I/O errors, &c), 2 to indicate a corrupt compressed file, 3 for an internal consistency error (eg, bug) which caused *bzip2* to panic.

## OPTIONS

### **-c --stdout**

Compress or decompress to standard output.

### **-d --decompress**

Force decompression. *bzip2*, *bunzip2* and *bzcat* are really the same program, and the decision about what actions to take is done on the basis of which name is used. This flag overrides that mechanism, and forces *bzip2* to decompress.

### **-z --compress**

The complement to *-d*: forces compression, regardless of the invocation name.

### **-t --test**

Check integrity of the specified file(s), but don't decompress them. This really performs a trial decompression and throws away the result.

### **-f --force**

Force overwrite of output files. Normally, *bzip2* will not overwrite existing output files. Also forces *bzip2* to break hard links to files, which it otherwise wouldn't do.

*bzip2* normally declines to decompress files which don't have the correct magic header bytes. If forced (*-f*), however, it will pass such files through unmodified. This is how GNU *gzip* behaves.

### **-k --keep**

Keep (don't delete) input files during compression or decompression.

### **-s --small**

Reduce memory usage, for compression, decompression and testing. Files are decompressed and tested using a modified algorithm which only requires 2.5 bytes per block byte. This means any file can be decompressed in 2300k of memory, albeit at about half the normal speed.

During compression, *-s* selects a block size of 200k, which limits memory use to around the same

figure, at the expense of your compression ratio. In short, if your machine is low on memory (8 megabytes or less), use `-s` for everything. See MEMORY MANAGEMENT below.

**-q --quiet**

Suppress non-essential warning messages. Messages pertaining to I/O errors and other critical events will not be suppressed.

**-v --verbose**

Verbose mode -- show the compression ratio for each file processed. Further `-v`'s increase the verbosity level, spewing out lots of information which is primarily of interest for diagnostic purposes.

**-L --license -V --version**

Display the software version, license terms and conditions.

**-1 (or --fast) to -9 (or --best)**

Set the block size to 100 k, 200 k .. 900 k when compressing. Has no effect when decompressing. See MEMORY MANAGEMENT below. The `--fast` and `--best` aliases are primarily for GNU gzip compatibility. In particular, `--fast` doesn't make things significantly faster. And `--best` merely selects the default behaviour.

**--**

Treats all subsequent arguments as file names, even if they start with a dash. This is so you can handle files with names beginning with a dash, for example: `bzip2 -- -myfilename`.

**--repetitive-fast --repetitive-best**

These flags are redundant in versions 0.9.5 and above. They provided some coarse control over the behaviour of the sorting algorithm in earlier versions, which was sometimes useful. 0.9.5 and above have an improved algorithm which renders these flags irrelevant.

## MEMORY MANAGEMENT

*bzip2* compresses large files in blocks. The block size affects both the compression ratio achieved, and the amount of memory needed for compression and decompression. The flags `-1` through `-9` specify the block size to be 100,000 bytes through 900,000 bytes (the default) respectively. At decompression time, the block size used for compression is read from the header of the compressed file, and *bunzip2* then allocates itself just enough memory to decompress the file. Since block sizes are stored in compressed files, it follows that the flags `-1` to `-9` are irrelevant to and so ignored during decompression.

Compression and decompression requirements, in bytes, can be estimated as:

Compression:  $400k + (8 \times \text{block size})$

Decompression:  $100k + (4 \times \text{block size})$ , or  
 $100k + (2.5 \times \text{block size})$

Larger block sizes give rapidly diminishing marginal returns. Most of the compression comes from the first two or three hundred k of block size, a fact worth bearing in mind when using *bzip2* on small machines. It is also important to appreciate that the decompression memory requirement is set at compression time by the choice of block size.

For files compressed with the default 900k block size, *bunzip2* will require about 3700 kbytes to decompress. To support decompression of any file on a 4 megabyte machine, *bunzip2* has an option to decompress using approximately half this amount of memory, about 2300 kbytes. Decompression speed is also halved, so you should use this option only where necessary. The relevant flag is `-s`.

In general, try and use the largest block size memory constraints allow, since that maximises the compression achieved. Compression and decompression speed are virtually unaffected by block size.

Another significant point applies to files which fit in a single block -- that means most files you'd encounter using a large block size. The amount of real memory touched is proportional to the size of the file, since

the file is smaller than a block. For example, compressing a file 20,000 bytes long with the flag -9 will cause the compressor to allocate around 7600k of memory, but only touch 400k + 20000 \* 8 = 560 kbytes of it. Similarly, the decompressor will allocate 3700k but only touch 100k + 20000 \* 4 = 180 kbytes.

Here is a table which summarises the maximum memory usage for different block sizes. Also recorded is the total compressed size for 14 files of the Calgary Text Compression Corpus totalling 3,141,622 bytes. This column gives some feel for how compression varies with block size. These figures tend to understate the advantage of larger block sizes for larger files, since the Corpus is dominated by smaller files.

| Flag | Compress<br>usage | Decompress<br>usage | Decompress<br>-s usage | Corpus<br>Size |
|------|-------------------|---------------------|------------------------|----------------|
| -1   | 1200k             | 500k                | 350k                   | 914704         |
| -2   | 2000k             | 900k                | 600k                   | 877703         |
| -3   | 2800k             | 1300k               | 850k                   | 860338         |
| -4   | 3600k             | 1700k               | 1100k                  | 846899         |
| -5   | 4400k             | 2100k               | 1350k                  | 845160         |
| -6   | 5200k             | 2500k               | 1600k                  | 838626         |
| -7   | 6100k             | 2900k               | 1850k                  | 834096         |
| -8   | 6800k             | 3300k               | 2100k                  | 828642         |
| -9   | 7600k             | 3700k               | 2350k                  | 828642         |

## RECOVERING DATA FROM DAMAGED FILES

*bzip2* compresses files in blocks, usually 900kbytes long. Each block is handled independently. If a media or transmission error causes a multi-block .bz2 file to become damaged, it may be possible to recover data from the undamaged blocks in the file.

The compressed representation of each block is delimited by a 48-bit pattern, which makes it possible to find the block boundaries with reasonable certainty. Each block also carries its own 32-bit CRC, so damaged blocks can be distinguished from undamaged ones.

*bzip2recover* is a simple program whose purpose is to search for blocks in .bz2 files, and write each block out into its own .bz2 file. You can then use *bzip2* -t to test the integrity of the resulting files, and decompress those which are undamaged.

*bzip2recover* takes a single argument, the name of the damaged file, and writes a number of files "rec00001file.bz2", "rec00002file.bz2", etc, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing -- for example, "bzip2 -dc rec\*file.bz2 > recovered\_data" -- processes the files in the correct order.

*bzip2recover* should be of most use dealing with large .bz2 files, as these will contain many blocks. It is clearly futile to use it on damaged single-block files, since a damaged block cannot be recovered. If you wish to minimise any potential data loss through media or transmission errors, you might consider compressing with a smaller block size.

## PERFORMANCE NOTES

The sorting phase of compression gathers together similar strings in the file. Because of this, files containing very long runs of repeated symbols, like "aabaabaabaab ..." (repeated several hundred times) may compress more slowly than normal. Versions 0.9.5 and above fare much better than previous versions in this respect. The ratio between worst-case and average-case compression time is in the region of 10:1. For previous versions, this figure was more like 100:1. You can use the -vvvv option to monitor progress in great detail, if you want.

Decompression speed is unaffected by these phenomena.

*bzip2* usually allocates several megabytes of memory to operate in, and then charges all over it in a fairly random fashion. This means that performance, both for compressing and decompressing, is largely determined by the speed at which your machine can service cache misses. Because of this, small changes to the code to reduce the miss rate have been observed to give disproportionately large performance improvements. I imagine *bzip2* will perform best on machines with very large caches.

## CAVEATS

I/O error messages are not as helpful as they could be. *bzip2* tries hard to detect I/O errors and exit cleanly, but the details of what the problem is sometimes seem rather misleading.

This manual page pertains to version 1.0.3 of *bzip2*. Compressed data created by this version is entirely forwards and backwards compatible with the previous public releases, versions 0.1pl2, 0.9.0, 0.9.5, 1.0.0, 1.0.1 and 1.0.2, but with the following exception: 0.9.0 and above can correctly decompress multiple concatenated compressed files. 0.1pl2 cannot do this; it will stop after decompressing just the first file in the stream.

*bzip2recover* versions prior to 1.0.2 used 32-bit integers to represent bit positions in compressed files, so they could not handle compressed files more than 512 megabytes long. Versions 1.0.2 and above use 64-bit ints on some platforms which support them (GNU supported targets, and Windows). To establish whether or not *bzip2recover* was built with such a limitation, run it without arguments. In any event you can build yourself an unlimited version if you can recompile it with `MaybeUInt64` set to be an unsigned 64-bit integer.

## AUTHOR

Julian Seward, [jsewardbzip.org](http://jsewardbzip.org).

<http://www.bzip.org>

The ideas embodied in *bzip2* are due to (at least) the following people: Michael Burrows and David Wheeler (for the block sorting transformation), David Wheeler (again, for the Huffman coder), Peter Fenwick (for the structured coding model in the original *bzip*, and many refinements), and Alistair Moffat, Radford Neal and Ian Witten (for the arithmetic coder in the original *bzip*). I am much indebted for their help, support and advice. See the manual in the source distribution for pointers to sources of documentation. Christian von Roques encouraged me to look for faster sorting algorithms, so as to speed up compression. Bela Lubkin encouraged me to improve the worst-case compression performance. Donna Robinson XMLised the documentation. The *bz\** scripts are derived from those of GNU *gzip*. Many people sent patches, helped with portability problems, lent machines, gave advice and were generally helpful.

**NAME**

busy – keep the terminal busy

**SYNOPSIS**

**edrc/bin/busy**

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

This command does nothing productive. It is used to keep a TCP/IP connection open. Using this command you will prevent a connection loss caused by a firewall which disconnects a session after a certain TCP timeout is reached.

To end type *Ctrl+C* .

**OPTIONS**

-

**ENVIRONMENT****\$BUSY\_TURNING\_INTERVAL**

define the speed of the rotating slash in seconds.

Sometimes the default speed of 1 second might not be feasible, for example when old output has to be checked in the terminal scrolling the window up, the output of **busy** will cause the terminal to jump back to the line where the **busy** slash is turning.

**\$BUSY\_ANIMATION**

set the animation to display. Possible settings are *inflate*, *spin*, *snip*, *snap*, *blink*, *cross*, *target*, *jump* and *slash*.

**EXIT STATUS**

**0** always.

**FILES**

-

**EXAMPLES**

-

**SEE ALSO**

**edrcintro(1)**, **edrc(1)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

busy was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net)

.

**COPYRIGHT**

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

**bwcreate** – create a BarbedWire file

**SYNOPSIS**

**edrc/bin/bwcreate** [ **-h** ]

**bwcreate** **-s** [ **-f** ] [ **-p** *profile* ] [ **-c** *config\_file* ] [ **-o** *outputfile* ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

Create a BarbedWire file that contains file properties and checksum of files on filesystems defined in the **bwcreate.cfg**(4) file. The BarbedWire file is used to track changes on (operating system) files.

The intention is to create a BarbedWire file for a system files in regular intervals. Doing this it is possible to track for files whose credentials changed. This might help to detect system intrusion or to find causes of system problems.

**bwcreate** saves all information in a compressed format to the BarbedWire file. The contents of the BarbedWire file can be listed using the **lscomp**(1), **llcomp**(1) and **catcomp**(1) commands.

**OPTIONS**

**-h** usage message.

**-s** start BarbedWire file creation

**-f** force overwriting of output file

**-l** list all profiles defined in the configuration file.

**-c** *config\_file*  
configuration file for **bwcreate**. Default configuration file is **edrc/etc/bwcreate.cfg**.

**-o** *output\_file*  
output filename, the BarbedWire file, without the **.bw** suffix.

**-p** *profile* Profile (=set of options) to be used to create the BarbedWire file. If this option is not specified the profile with the name *default* is used.

## FILE FORMAT

The BarbedWire file version 100 consists of a **HEADER** and a **FILELIST** section. Use **lscomp(1)**, **llcomp(1)** and **catcomp(1)** to display the contents of a BarbedWire ( **.bw** ) file.

### HEADER

The header of the BarbedWire file consists of the entries explained below. The fileformat is **OPTION=VALUE** . Between the **OPTION**, the = and the **VALUE** are no spaces. Each **OPTION** is on a separate line. The sequence of the **OPTION** entries is as explained below, but can change without notice. Therefore, if you write programs that resolve the **OPTION** do not rely on the sequence.

### APPLICATION

Name of the application that created the file. This setting is set to *BarbedWire*.

### FILEVERSION

Version of the BarbedWire file format.

### HOSTNAME

Hostname where the BarbedWire file has been created.

**PROFILE** Profile specified with the **-p** option.

### DESCRIPTION

Description text of the profile.

### CREATED

Date and time when the BarbedWire file has been created. The Format is: YYYYMMDDhh-mmss.

### CREATED\_BY

Command that created the BarbedWire file. This setting is set to *bwcreate*.

### ENVIRONMENT

Server environment where the BarbedWire file has been created. This setting is resolved by **bwcreate** using the **server\_environment** command.

### ENVIRONMENT\_DESCRIPTION

Detail description of the **ENVIRONMENT**. This setting is resolved by **bwcreate** using the **server\_environment -d** command.

**OSID** Osid, see **osid(3)**, of the system where the BarbedWire file has been created.

**UNAME** System information of the system where the BarbedWire file has been created. This setting equals to the **uname -a** command output.

**USER** User that created the BarbedWire file.



**CONFIGFILE**

Configuration file used to create the BarbedWire file.

**FILESYSTEMS**

List of filesystems whose files are resolved and part of the **FILELIST**.

**OUTPUT\_DIRECTORY**

Output directory where the BarbedWire file has been written to during creation.

**OUTPUT\_FILE**

Output file name (BarbedWire file name) during creation.

**FILELIST**

The file list section of the Barbed Wire file consists of the lines explained below. The first line in the **FILELIST** section is a header listing all fields that are present for all files in the filelist separated by a ; . All following lines are file entries. The filelist header defines the sequence and the name of the fields. Therefore, if you write programs that resolve the **FILELIST** of a BarbedWire file use the filelist header entry to resolve the fields of the filelist entries and do not rely on the sequence documented here, due to the fact that this sequence can change without notice.

**TYPE** File type:

**f** file

**d**  
directory

**c** Character oriented device file (special file)

**b**  
Block oriented device file (special file)

**PERM\_OWNER**

Symbolic permission representation for the file owner. See **chmod(1)** for more information.

**PERM\_GROUP**

Symbolic permission representation for the file group. See **chmod(1)** for more information.

**PERM\_OTHER**

Symbolic permission representation for the others file permissions. See **chmod(1)** or **ls(1)** for more information.

**CHECKSUM**

MD5 checksum of the file. See **md5sum(1)** for more information.

**INODE**

Inode number of the file. See **ls(1)** or **stat(1)** for more information.

**BLOCKS**

Number of blocks on the filesystem used by the file. See **ls(1)** or **stat(1)** for more information.

**PERM\_ALL**

Complete file permission settings of the file. See **chmod**(1) or **ls**(1) for more information.

**REF**

Number of references to a file. See **ls**(1) or **stat**(1) for more information.

**OWNER**

Owner of the file. See **chown**(1) or **ls**(1) for more information.

**GROUP**

Group of the file. See **chgrp**(1) or **ls**(1) for more information.

**BITES**

Size of the file in bytes. See **ls**(1) or **stat**(1) for more information.

**NODE**

Node of the file. This applies to special files only. See **mknod**(1) or **ls**(1) for more information.

**MTIME**

Modification time of the file. See **ls**(1) or **stat**(1) for more information.

**FILE**

File name.

**ENVIRONMENT**

-

**EXIT STATUS**

- 0** no error.
- 2** operating system is not supported. See **osid**(3) if you get this error.
- 3** profile specified in the **-p** option is not defined in the configuration file.
- 4** usage listed.
- 5** **bwcreate** has been aborted pressing Ctrl+C or by issuing **kill bwcreate\_pid**.
- 6** configuration file **edrc/etc/bwcreate.cfg** does not exist.
- 7** output file already exists.
- 8** cannot write to the logfile.

**FILES****edrc/etc/bwcreate.cfg**

default configuration file for **bwcreate**.

**edrc/var/log/**

default logfile location of **bwcreate**.

**edrc/var/barbedwire/cache/bwcreate.duration**

record of durations of past runs of **bwcreate** to calculate the completion estimation.

**.bw**

BarbedWire output file suffix. The content of a BarbedWire file can be listed using **lscomp**, **llcomp** and **catcomp**.

**EXAMPLES****1) display contents of a BarbedWire file****Display contents:**

```
[/dat/tmp]
[root@rh7mzv7t001][*edrc*/bash]: lscomp rh7mzv7t001.bw
HEADER
FILELIST

[/dat/tmp]
[root@rh7mzv7t001][*edrc*/bash]: llcomp rh7mzv7t001.bw

root 2007-03-11 12:41:07 HEADER
root 2007-03-11 12:41:07 FILELIST
```

**Display header:**

```
[/dat/tmp]
[root@rh7mzv7t001][*edrc*/bash]: catcomp rh7mzv7t001.bw HEADER

APPLICATION=BarbedWire
FILEVERSION=100
HOSTNAME=rh7mzv7t001
PROFILE=dvlp_serial
DESCRIPTION=Development (serial execution)
CREATED=20070311124107
CREATED_BY=bwcreate
ENVIRONMENT=EDRC_DVLP
ENVIRONMENT_DESCRIPTION=Beringen, EDRC Development
OSID=Linux
UNAME=Linux rh7mzv7t001 2.4.7-10 #1 Thu Sep 6 17:27:27 EDT 2001 i686 unkn
USER=root
CONFIGFILE=/opt/edrc/etc/bwcreate.cfg
FILESYSTEMS=/etc/gnome /tmp /usr/local/bin /usr/local/lib
OUTPUT_DIRECTORY=/dat/tmp
OUTPUT_FILE=rh7mzv7t001_20070311124107.bw
```

**Display filelist:**

```
[/dat/tmp]
[root@rh7mzv7t001][*edrc*/bash]: catcomp rh7mzv7t001.bw FILELIST
```

```

TYPE;PERM_OWNER;PERM_GROUP;PERM_OTHER;CHECKSUM;INODE;BLOCKS;PERM_ALL;REF;
d;rwX;r-x;r-x;;114135;4;drwxr-xr-x;4;root;root;4096;;2004-05-05 12:46:48;
d;rwX;r-x;r-x;;241414;4;drwxr-xr-x;2;root;root;4096;;2004-05-05 12:45:53;
f;rw-;r--;r--;7afdb668ef553f5176736ff5081e3f13;241462;32;-rw-r--r--;1;roo
:
:

```

## 2) formatted display of BarbedWire filelist

```

[/dat/tmp]
[root@rh7mzv7t001][*edrc*/bash]: catcomp rh7mzv7t001.bw FILELIST |\
 print_list | more

TYPE PERM_OWNER PERM_GROUP PERM_OTHER CHECKSUM >>
---- -
d rwX r-x r-x >>
d rwX r-x r-x >>
f rw- r-- r-- 7afdb668ef553f5176736ff5081e>>
--More--

```

## SEE ALSO

**bwcompare(1)**, **bwcreate.cfg(4)**, **catcomp(1)**, **chgrp(1)**, **chmod(1)**, **edrcintro(1)**, **ls(1)**, **lscmp(1)**, **llcomp(1)**, **mknod(1)**, **more(1)**, **print\_list(3)**, **stat(1)**

## NOTES

-

## BUGS

-

## AUTHOR

bwcreate was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

## COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

bwcreate.cfg – configuration file for bwcreate

**SYNOPSIS**

edrc/etc/bwcreate.cfg

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

This is the configuration file for the **bwcreate** command.

**FILEFORMAT**

The format for options is **OPTION=VALUE** .

The format of a profile definition is:

```
PROFILE_name(){
 OPTION1=VALUE
 OPTION2=VALUE
 :
 OPTIONn=VALUE
}
```

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION**. If you like to use default settings simply do not specify a *VALUE*.

Normally there are multiple profiles defined in one configuration file.

**OPTIONS****GLOBAL OPTIONS**

This options influence the general behavior of the **bwcreate** command.

**LOG**      Log output dir of **bwcreate**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

## PROFILE OPTIONS

A profile has the following structure:

```
PROFILE_name(){
 DESCRIPTION='Text'
 FILESYSTEMS=" filesystems"
 OUTPUT_DIRECTORY=directory
 OUTPUT_FILE=filename
 OUTPUT_FILE_PERMISSIONS=perm
 PARALLEL_EXECUTION=True|False
}
```

The options recognized in a profile are:

*name* Name of the profile. The name defined here has to be specified in the **-p** option in the **bwcreate** command. The profile with the name *default* is called when the **-p** option is not specified when using the **bwcreate** command. Therefore each **bwcreate.cfg** file should at least define a profile with the name *default*.

## DESCRIPTION

Profile description. The description text has to be enclosed in *\*single\** quotes.

Example: DESCRIPTION='common Unix/Linux'

Default: DESCRIPTION=""

## FILESYSTEMS

Filesystems to be listed in the BarbedWire file. It is also possible to define directories within a filesystem. If there are extra filesystems for **/opt**, **/usr** it is not sufficient to list only the root filesystem / here, due to the fact that the file search is local to a filesystem.

Example: FILESYSTEMS="/ /var /usr /opt"

Default: FILESYSTEMS=""

## OUTPUT\_DIRECTORY

Existing directory where the BarbedWire file is created. This directory should have at least twice the space available as the resulting BarbedWire file size.

Example: OUTPUT\_DIRECTORY=/dat/report/barbedwire

Default: OUTPUT\_DIRECTORY=/var/tmp

**OUTPUT\_FILE**

Filename of BarbedWire file in OUTPUT\_DIRECTORY without suffix. A BarbedWire file has the suffix **.bw**.

Example: OUTPUT\_FILE=\$HOSTNAME\_\$NOW

Default: OUTPUT\_FILE=barbedwire\_outfile\_\$HOSTNAME\_\$TODAY

**OUTPUT\_FILE\_PERMISSIONS**

File permissions of BarbedWire file produced.

Example: OUTPUT\_FILE\_PERMISSIONS=400

Default: OUTPUT\_FILE\_PERMISSIONS=444

**PARALLEL\_EXECUTION**

Execute the resolution of the fileproperties for all filesystems or directories listed in **FILESYSTEMS** in parallel.

Example: PARALLEL\_EXECUTION=False

Default: PARALLEL\_EXECUTION=True

**VARIABLES**

The following variables can be used in profiles. Do not use other variables in a **bwcreate.cfg** file seen by **bwcreate** due to the fact that the existence and/or contents can change without notice.

**\$HOSTNAME**

The hostname of the system where the **bwcreate** command is executed.

**\$SERVER\_ENVIRONMENT**

The name of the server environment as returned by the **server\_environment** command. See also **server\_environment(3)** for more information about this command.

**\$TODAY** Date in the format YYYYMMDD (e.g. 20061206).

**\$NOW** Date and time in the format YYYYMMDDHHmm (e.g. 200612062307).

**SEE ALSO**

**bwcreate(1)**, **edrcintro(1)**, **server\_environment(3)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

bwcreate.cfg was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.



**NAME**

**bzip2**, **bunzip2** – a block-sorting file compressor, v1.0.3  
**bzcat** – decompresses files to stdout  
**bzip2recover** – recovers data from damaged bzip2 files

**SYNOPSIS**

**bzip2** [ **-cdfkqstvwVL123456789** ] [ *filenames ...* ]  
**bunzip2** [ **-fkvsVL** ] [ *filenames ...* ]  
**bzcat** [ **-s** ] [ *filenames ...* ]  
**bzip2recover** *filename*

**DESCRIPTION**

*bzip2* compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM family of statistical compressors.

The command-line options are deliberately very similar to those of *GNU gzip*, but they are not identical.

*bzip2* expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name "original\_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

*bzip2* and *bunzip2* will by default not overwrite existing files. If you want this to happen, specify the **-f** flag.

If no file names are specified, *bzip2* compresses from standard input to standard output. In this case, *bzip2* will decline to write compressed output to a terminal, as this would be entirely incomprehensible and therefore pointless.

*bunzip2* (or *bzip2 -d*) decompresses all specified files. Files which were not created by *bzip2* will be detected and ignored, and a warning issued. *bzip2* attempts to guess the filename for the decompressed file from that of the compressed file as follows:

```
filename.bz2 becomes filename
filename.bz becomes filename
filename.tbz2 becomes filename.tar
filename.tbz becomes filename.tar
anyothername becomes anyothername.out
```

If the file does not end in one of the recognised endings, *.bz2*, *.bz*, *.tbz2* or *.tbz*, *bzip2* complains that it cannot guess the name of the original file, and uses the original name with *.out* appended.

As with compression, supplying no filenames causes decompression from standard input to standard output.

*bunzip2* will correctly decompress a file which is the concatenation of two or more compressed files. The result is the concatenation of the corresponding uncompressed files. Integrity testing (**-t**) of concatenated compressed files is also supported.

You can also compress or decompress files to the standard output by giving the **-c** flag. Multiple files may be compressed and decompressed like this. The resulting outputs are fed sequentially to stdout.

Compression of multiple files in this manner generates a stream containing multiple compressed file representations. Such a stream can be decompressed correctly only by *bzip2* version 0.9.0 or later. Earlier versions of *bzip2* will stop after decompressing the first file in the stream.

*bzcat* (or *bzip2 -dc*) decompresses all specified files to the standard output.

*bzip2* will read arguments from the environment variables *BZIP2* and *BZIP*, in that order, and will process them before any arguments read from the command line. This gives a convenient way to supply default arguments.

Compression is always performed, even if the compressed file is slightly larger than the original. Files of less than about one hundred bytes tend to get larger, since the compression mechanism has a constant overhead in the region of 50 bytes. Random data (including the output of most file compressors) is coded at about 8.05 bits per byte, giving an expansion of around 0.5%.

As a self-check for your protection, *bzip2* uses 32-bit CRCs to make sure that the decompressed version of a file is identical to the original. This guards against corruption of the compressed data, and against undetected bugs in *bzip2* (hopefully very unlikely). The chances of data corruption going undetected is microscopic, about one chance in four billion for each file processed. Be aware, though, that the check occurs upon decompression, so it can only tell you that something is wrong. It can't help you recover the original uncompressed data. You can use *bzip2recover* to try to recover data from damaged files.

Return values: 0 for a normal exit, 1 for environmental problems (file not found, invalid flags, I/O errors, &c), 2 to indicate a corrupt compressed file, 3 for an internal consistency error (eg, bug) which caused *bzip2* to panic.

## OPTIONS

### **-c --stdout**

Compress or decompress to standard output.

### **-d --decompress**

Force decompression. *bzip2*, *bunzip2* and *bzcat* are really the same program, and the decision about what actions to take is done on the basis of which name is used. This flag overrides that mechanism, and forces *bzip2* to decompress.

### **-z --compress**

The complement to *-d*: forces compression, regardless of the invocation name.

### **-t --test**

Check integrity of the specified file(s), but don't decompress them. This really performs a trial decompression and throws away the result.

### **-f --force**

Force overwrite of output files. Normally, *bzip2* will not overwrite existing output files. Also forces *bzip2* to break hard links to files, which it otherwise wouldn't do.

*bzip2* normally declines to decompress files which don't have the correct magic header bytes. If forced (*-f*), however, it will pass such files through unmodified. This is how GNU *gzip* behaves.

### **-k --keep**

Keep (don't delete) input files during compression or decompression.

### **-s --small**

Reduce memory usage, for compression, decompression and testing. Files are decompressed and tested using a modified algorithm which only requires 2.5 bytes per block byte. This means any file can be decompressed in 2300k of memory, albeit at about half the normal speed.

During compression, *-s* selects a block size of 200k, which limits memory use to around the same

figure, at the expense of your compression ratio. In short, if your machine is low on memory (8 megabytes or less), use `-s` for everything. See MEMORY MANAGEMENT below.

**-q --quiet**

Suppress non-essential warning messages. Messages pertaining to I/O errors and other critical events will not be suppressed.

**-v --verbose**

Verbose mode -- show the compression ratio for each file processed. Further `-v`'s increase the verbosity level, spewing out lots of information which is primarily of interest for diagnostic purposes.

**-L --license -V --version**

Display the software version, license terms and conditions.

**-1 (or --fast) to -9 (or --best)**

Set the block size to 100 k, 200 k .. 900 k when compressing. Has no effect when decompressing. See MEMORY MANAGEMENT below. The `--fast` and `--best` aliases are primarily for GNU gzip compatibility. In particular, `--fast` doesn't make things significantly faster. And `--best` merely selects the default behaviour.

**--**

Treats all subsequent arguments as file names, even if they start with a dash. This is so you can handle files with names beginning with a dash, for example: `bzip2 -- -myfilename`.

**--repetitive-fast --repetitive-best**

These flags are redundant in versions 0.9.5 and above. They provided some coarse control over the behaviour of the sorting algorithm in earlier versions, which was sometimes useful. 0.9.5 and above have an improved algorithm which renders these flags irrelevant.

## MEMORY MANAGEMENT

*bzip2* compresses large files in blocks. The block size affects both the compression ratio achieved, and the amount of memory needed for compression and decompression. The flags `-1` through `-9` specify the block size to be 100,000 bytes through 900,000 bytes (the default) respectively. At decompression time, the block size used for compression is read from the header of the compressed file, and *bunzip2* then allocates itself just enough memory to decompress the file. Since block sizes are stored in compressed files, it follows that the flags `-1` to `-9` are irrelevant to and so ignored during decompression.

Compression and decompression requirements, in bytes, can be estimated as:

Compression:  $400k + (8 \times \text{block size})$

Decompression:  $100k + (4 \times \text{block size})$ , or  
 $100k + (2.5 \times \text{block size})$

Larger block sizes give rapidly diminishing marginal returns. Most of the compression comes from the first two or three hundred k of block size, a fact worth bearing in mind when using *bzip2* on small machines. It is also important to appreciate that the decompression memory requirement is set at compression time by the choice of block size.

For files compressed with the default 900k block size, *bunzip2* will require about 3700 kbytes to decompress. To support decompression of any file on a 4 megabyte machine, *bunzip2* has an option to decompress using approximately half this amount of memory, about 2300 kbytes. Decompression speed is also halved, so you should use this option only where necessary. The relevant flag is `-s`.

In general, try and use the largest block size memory constraints allow, since that maximises the compression achieved. Compression and decompression speed are virtually unaffected by block size.

Another significant point applies to files which fit in a single block -- that means most files you'd encounter using a large block size. The amount of real memory touched is proportional to the size of the file, since

the file is smaller than a block. For example, compressing a file 20,000 bytes long with the flag -9 will cause the compressor to allocate around 7600k of memory, but only touch 400k + 20000 \* 8 = 560 kbytes of it. Similarly, the decompressor will allocate 3700k but only touch 100k + 20000 \* 4 = 180 kbytes.

Here is a table which summarises the maximum memory usage for different block sizes. Also recorded is the total compressed size for 14 files of the Calgary Text Compression Corpus totalling 3,141,622 bytes. This column gives some feel for how compression varies with block size. These figures tend to understate the advantage of larger block sizes for larger files, since the Corpus is dominated by smaller files.

| Flag | Compress<br>usage | Decompress<br>usage | Decompress<br>-s usage | Corpus<br>Size |
|------|-------------------|---------------------|------------------------|----------------|
| -1   | 1200k             | 500k                | 350k                   | 914704         |
| -2   | 2000k             | 900k                | 600k                   | 877703         |
| -3   | 2800k             | 1300k               | 850k                   | 860338         |
| -4   | 3600k             | 1700k               | 1100k                  | 846899         |
| -5   | 4400k             | 2100k               | 1350k                  | 845160         |
| -6   | 5200k             | 2500k               | 1600k                  | 838626         |
| -7   | 6100k             | 2900k               | 1850k                  | 834096         |
| -8   | 6800k             | 3300k               | 2100k                  | 828642         |
| -9   | 7600k             | 3700k               | 2350k                  | 828642         |

## RECOVERING DATA FROM DAMAGED FILES

*bzip2* compresses files in blocks, usually 900kbytes long. Each block is handled independently. If a media or transmission error causes a multi-block .bz2 file to become damaged, it may be possible to recover data from the undamaged blocks in the file.

The compressed representation of each block is delimited by a 48-bit pattern, which makes it possible to find the block boundaries with reasonable certainty. Each block also carries its own 32-bit CRC, so damaged blocks can be distinguished from undamaged ones.

*bzip2recover* is a simple program whose purpose is to search for blocks in .bz2 files, and write each block out into its own .bz2 file. You can then use *bzip2* -t to test the integrity of the resulting files, and decompress those which are undamaged.

*bzip2recover* takes a single argument, the name of the damaged file, and writes a number of files "rec00001file.bz2", "rec00002file.bz2", etc, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing -- for example, "bzip2 -dc rec\*file.bz2 > recovered\_data" -- processes the files in the correct order.

*bzip2recover* should be of most use dealing with large .bz2 files, as these will contain many blocks. It is clearly futile to use it on damaged single-block files, since a damaged block cannot be recovered. If you wish to minimise any potential data loss through media or transmission errors, you might consider compressing with a smaller block size.

## PERFORMANCE NOTES

The sorting phase of compression gathers together similar strings in the file. Because of this, files containing very long runs of repeated symbols, like "aabaabaabaab ..." (repeated several hundred times) may compress more slowly than normal. Versions 0.9.5 and above fare much better than previous versions in this respect. The ratio between worst-case and average-case compression time is in the region of 10:1. For previous versions, this figure was more like 100:1. You can use the -vvvv option to monitor progress in great detail, if you want.

Decompression speed is unaffected by these phenomena.

*bzip2* usually allocates several megabytes of memory to operate in, and then charges all over it in a fairly random fashion. This means that performance, both for compressing and decompressing, is largely determined by the speed at which your machine can service cache misses. Because of this, small changes to the code to reduce the miss rate have been observed to give disproportionately large performance improvements. I imagine *bzip2* will perform best on machines with very large caches.

## CAVEATS

I/O error messages are not as helpful as they could be. *bzip2* tries hard to detect I/O errors and exit cleanly, but the details of what the problem is sometimes seem rather misleading.

This manual page pertains to version 1.0.3 of *bzip2*. Compressed data created by this version is entirely forwards and backwards compatible with the previous public releases, versions 0.1pl2, 0.9.0, 0.9.5, 1.0.0, 1.0.1 and 1.0.2, but with the following exception: 0.9.0 and above can correctly decompress multiple concatenated compressed files. 0.1pl2 cannot do this; it will stop after decompressing just the first file in the stream.

*bzip2recover* versions prior to 1.0.2 used 32-bit integers to represent bit positions in compressed files, so they could not handle compressed files more than 512 megabytes long. Versions 1.0.2 and above use 64-bit ints on some platforms which support them (GNU supported targets, and Windows). To establish whether or not *bzip2recover* was built with such a limitation, run it without arguments. In any event you can build yourself an unlimited version if you can recompile it with `MaybeUInt64` set to be an unsigned 64-bit integer.

## AUTHOR

Julian Seward, [jsewardbzip.org](http://jsewardbzip.org).

<http://www.bzip.org>

The ideas embodied in *bzip2* are due to (at least) the following people: Michael Burrows and David Wheeler (for the block sorting transformation), David Wheeler (again, for the Huffman coder), Peter Fenwick (for the structured coding model in the original *bzip*, and many refinements), and Alistair Moffat, Radford Neal and Ian Witten (for the arithmetic coder in the original *bzip*). I am much indebted for their help, support and advice. See the manual in the source distribution for pointers to sources of documentation. Christian von Roques encouraged me to look for faster sorting algorithms, so as to speed up compression. Bela Lubkin encouraged me to improve the worst-case compression performance. Donna Robinson XMLised the documentation. The *bz\** scripts are derived from those of GNU *gzip*. Many people sent patches, helped with portability problems, lent machines, gave advice and were generally helpful.

**NAME**

**bzcmp**, **bzdiff** – compare bzip2 compressed files

**SYNOPSIS**

**bzcmp** [ *cmp\_options* ] *file1* [ *file2* ]

**bzdiff** [ *diff\_options* ] *file1* [ *file2* ]

**DESCRIPTION**

*Bzcmp* and *bzdiff* are used to invoke the *cmp* or the *diff* program on bzip2 compressed files. All options specified are passed directly to *cmp* or *diff*. If only 1 file is specified, then the files compared are *file1* and an uncompressed *file1.bz2*. If two files are specified, then they are uncompressed if necessary and fed to *cmp* or *diff*. The exit status from *cmp* or *diff* is preserved.

**SEE ALSO**

*cmp*(1), *diff*(1), *bzmore*(1), *bzless*(1), *bzgrep*(1), *bzip2*(1)

**BUGS**

Messages from the *cmp* or *diff* programs refer to temporary filenames instead of those specified.

**NAME**

**bzgrep**, **bzfgrep**, **bzegrep** – search possibly bzip2 compressed files for a regular expression

**SYNOPSIS**

**bzgrep** [ *grep\_options* ] [ **-e** ] *pattern filename...*

**bzegrep** [ *egrep\_options* ] [ **-e** ] *pattern filename...*

**bzfgrep** [ *fgrep\_options* ] [ **-e** ] *pattern filename...*

**DESCRIPTION**

*Bzgrep* is used to invoke the *grep* on bzip2-compressed files. All options specified are passed directly to *grep*. If no file is specified, then the standard input is decompressed if necessary and fed to *grep*. Otherwise the given files are uncompressed if necessary and fed to *grep*.

If *bzgrep* is invoked as *bzegrep* or *bzfgrep* then *egrep* or *fgrep* is used instead of *grep*. If the GREP environment variable is set, *bzgrep* uses it as the *grep* program to be invoked. For example:

for sh: GREP=fgrep bzgrep string files

for csh: (setenv GREP fgrep; bzgrep string files)

**AUTHOR**

Charles Levert (charles@comm.polymtl.ca). Adapted to bzip2 by Philippe Troin <phil@fifi.org> for Debian GNU/Linux.

**SEE ALSO**

*grep*(1), *egrep*(1), *fgrep*(1), *bzdiff*(1), *bzmore*(1), *bzless*(1), *bzip2*(1)

**NAME**

**bzip2**, **bunzip2** – a block-sorting file compressor, v1.0.3  
**bzcat** – decompresses files to stdout  
**bzip2recover** – recovers data from damaged bzip2 files

**SYNOPSIS**

**bzip2** [ **-cdfkqstvwVL123456789** ] [ *filenames ...* ]  
**bunzip2** [ **-fkvsVL** ] [ *filenames ...* ]  
**bzcat** [ **-s** ] [ *filenames ...* ]  
**bzip2recover** *filename*

**DESCRIPTION**

*bzip2* compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM family of statistical compressors.

The command-line options are deliberately very similar to those of *GNU gzip*, but they are not identical.

*bzip2* expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name "original\_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

*bzip2* and *bunzip2* will by default not overwrite existing files. If you want this to happen, specify the **-f** flag.

If no file names are specified, *bzip2* compresses from standard input to standard output. In this case, *bzip2* will decline to write compressed output to a terminal, as this would be entirely incomprehensible and therefore pointless.

*bunzip2* (or *bzip2 -d*) decompresses all specified files. Files which were not created by *bzip2* will be detected and ignored, and a warning issued. *bzip2* attempts to guess the filename for the decompressed file from that of the compressed file as follows:

```
filename.bz2 becomes filename
filename.bz becomes filename
filename.tbz2 becomes filename.tar
filename.tbz becomes filename.tar
anyothername becomes anyothername.out
```

If the file does not end in one of the recognised endings, *.bz2*, *.bz*, *.tbz2* or *.tbz*, *bzip2* complains that it cannot guess the name of the original file, and uses the original name with *.out* appended.

As with compression, supplying no filenames causes decompression from standard input to standard output.

*bunzip2* will correctly decompress a file which is the concatenation of two or more compressed files. The result is the concatenation of the corresponding uncompressed files. Integrity testing (**-t**) of concatenated compressed files is also supported.

You can also compress or decompress files to the standard output by giving the **-c** flag. Multiple files may be compressed and decompressed like this. The resulting outputs are fed sequentially to stdout.



Compression of multiple files in this manner generates a stream containing multiple compressed file representations. Such a stream can be decompressed correctly only by *bzip2* version 0.9.0 or later. Earlier versions of *bzip2* will stop after decompressing the first file in the stream.

*bzcat* (or *bzip2 -dc*) decompresses all specified files to the standard output.

*bzip2* will read arguments from the environment variables *BZIP2* and *BZIP*, in that order, and will process them before any arguments read from the command line. This gives a convenient way to supply default arguments.

Compression is always performed, even if the compressed file is slightly larger than the original. Files of less than about one hundred bytes tend to get larger, since the compression mechanism has a constant overhead in the region of 50 bytes. Random data (including the output of most file compressors) is coded at about 8.05 bits per byte, giving an expansion of around 0.5%.

As a self-check for your protection, *bzip2* uses 32-bit CRCs to make sure that the decompressed version of a file is identical to the original. This guards against corruption of the compressed data, and against undetected bugs in *bzip2* (hopefully very unlikely). The chances of data corruption going undetected is microscopic, about one chance in four billion for each file processed. Be aware, though, that the check occurs upon decompression, so it can only tell you that something is wrong. It can't help you recover the original uncompressed data. You can use *bzip2recover* to try to recover data from damaged files.

Return values: 0 for a normal exit, 1 for environmental problems (file not found, invalid flags, I/O errors, &c), 2 to indicate a corrupt compressed file, 3 for an internal consistency error (eg, bug) which caused *bzip2* to panic.

## OPTIONS

### **-c --stdout**

Compress or decompress to standard output.

### **-d --decompress**

Force decompression. *bzip2*, *bunzip2* and *bzcat* are really the same program, and the decision about what actions to take is done on the basis of which name is used. This flag overrides that mechanism, and forces *bzip2* to decompress.

### **-z --compress**

The complement to *-d*: forces compression, regardless of the invocation name.

### **-t --test**

Check integrity of the specified file(s), but don't decompress them. This really performs a trial decompression and throws away the result.

### **-f --force**

Force overwrite of output files. Normally, *bzip2* will not overwrite existing output files. Also forces *bzip2* to break hard links to files, which it otherwise wouldn't do.

*bzip2* normally declines to decompress files which don't have the correct magic header bytes. If forced (*-f*), however, it will pass such files through unmodified. This is how GNU *gzip* behaves.

### **-k --keep**

Keep (don't delete) input files during compression or decompression.

### **-s --small**

Reduce memory usage, for compression, decompression and testing. Files are decompressed and tested using a modified algorithm which only requires 2.5 bytes per block byte. This means any file can be decompressed in 2300k of memory, albeit at about half the normal speed.

During compression, *-s* selects a block size of 200k, which limits memory use to around the same

figure, at the expense of your compression ratio. In short, if your machine is low on memory (8 megabytes or less), use `-s` for everything. See MEMORY MANAGEMENT below.

**-q --quiet**

Suppress non-essential warning messages. Messages pertaining to I/O errors and other critical events will not be suppressed.

**-v --verbose**

Verbose mode -- show the compression ratio for each file processed. Further `-v`'s increase the verbosity level, spewing out lots of information which is primarily of interest for diagnostic purposes.

**-L --license -V --version**

Display the software version, license terms and conditions.

**-1 (or --fast) to -9 (or --best)**

Set the block size to 100 k, 200 k .. 900 k when compressing. Has no effect when decompressing. See MEMORY MANAGEMENT below. The `--fast` and `--best` aliases are primarily for GNU gzip compatibility. In particular, `--fast` doesn't make things significantly faster. And `--best` merely selects the default behaviour.

**--**

Treats all subsequent arguments as file names, even if they start with a dash. This is so you can handle files with names beginning with a dash, for example: `bzip2 -- -myfilename`.

**--repetitive-fast --repetitive-best**

These flags are redundant in versions 0.9.5 and above. They provided some coarse control over the behaviour of the sorting algorithm in earlier versions, which was sometimes useful. 0.9.5 and above have an improved algorithm which renders these flags irrelevant.

## MEMORY MANAGEMENT

*bzip2* compresses large files in blocks. The block size affects both the compression ratio achieved, and the amount of memory needed for compression and decompression. The flags `-1` through `-9` specify the block size to be 100,000 bytes through 900,000 bytes (the default) respectively. At decompression time, the block size used for compression is read from the header of the compressed file, and *bunzip2* then allocates itself just enough memory to decompress the file. Since block sizes are stored in compressed files, it follows that the flags `-1` to `-9` are irrelevant to and so ignored during decompression.

Compression and decompression requirements, in bytes, can be estimated as:

Compression:  $400k + (8 \times \text{block size})$

Decompression:  $100k + (4 \times \text{block size})$ , or  
 $100k + (2.5 \times \text{block size})$

Larger block sizes give rapidly diminishing marginal returns. Most of the compression comes from the first two or three hundred k of block size, a fact worth bearing in mind when using *bzip2* on small machines. It is also important to appreciate that the decompression memory requirement is set at compression time by the choice of block size.

For files compressed with the default 900k block size, *bunzip2* will require about 3700 kbytes to decompress. To support decompression of any file on a 4 megabyte machine, *bunzip2* has an option to decompress using approximately half this amount of memory, about 2300 kbytes. Decompression speed is also halved, so you should use this option only where necessary. The relevant flag is `-s`.

In general, try and use the largest block size memory constraints allow, since that maximises the compression achieved. Compression and decompression speed are virtually unaffected by block size.

Another significant point applies to files which fit in a single block -- that means most files you'd encounter using a large block size. The amount of real memory touched is proportional to the size of the file, since

the file is smaller than a block. For example, compressing a file 20,000 bytes long with the flag -9 will cause the compressor to allocate around 7600k of memory, but only touch 400k + 20000 \* 8 = 560 kbytes of it. Similarly, the decompressor will allocate 3700k but only touch 100k + 20000 \* 4 = 180 kbytes.

Here is a table which summarises the maximum memory usage for different block sizes. Also recorded is the total compressed size for 14 files of the Calgary Text Compression Corpus totalling 3,141,622 bytes. This column gives some feel for how compression varies with block size. These figures tend to understate the advantage of larger block sizes for larger files, since the Corpus is dominated by smaller files.

| Flag | Compress<br>usage | Decompress<br>usage | Decompress<br>-s usage | Corpus<br>Size |
|------|-------------------|---------------------|------------------------|----------------|
| -1   | 1200k             | 500k                | 350k                   | 914704         |
| -2   | 2000k             | 900k                | 600k                   | 877703         |
| -3   | 2800k             | 1300k               | 850k                   | 860338         |
| -4   | 3600k             | 1700k               | 1100k                  | 846899         |
| -5   | 4400k             | 2100k               | 1350k                  | 845160         |
| -6   | 5200k             | 2500k               | 1600k                  | 838626         |
| -7   | 6100k             | 2900k               | 1850k                  | 834096         |
| -8   | 6800k             | 3300k               | 2100k                  | 828642         |
| -9   | 7600k             | 3700k               | 2350k                  | 828642         |

## RECOVERING DATA FROM DAMAGED FILES

*bzip2* compresses files in blocks, usually 900kbytes long. Each block is handled independently. If a media or transmission error causes a multi-block .bz2 file to become damaged, it may be possible to recover data from the undamaged blocks in the file.

The compressed representation of each block is delimited by a 48-bit pattern, which makes it possible to find the block boundaries with reasonable certainty. Each block also carries its own 32-bit CRC, so damaged blocks can be distinguished from undamaged ones.

*bzip2recover* is a simple program whose purpose is to search for blocks in .bz2 files, and write each block out into its own .bz2 file. You can then use *bzip2* -t to test the integrity of the resulting files, and decompress those which are undamaged.

*bzip2recover* takes a single argument, the name of the damaged file, and writes a number of files "rec00001file.bz2", "rec00002file.bz2", etc, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing -- for example, "bzip2 -dc rec\*file.bz2 > recovered\_data" -- processes the files in the correct order.

*bzip2recover* should be of most use dealing with large .bz2 files, as these will contain many blocks. It is clearly futile to use it on damaged single-block files, since a damaged block cannot be recovered. If you wish to minimise any potential data loss through media or transmission errors, you might consider compressing with a smaller block size.

## PERFORMANCE NOTES

The sorting phase of compression gathers together similar strings in the file. Because of this, files containing very long runs of repeated symbols, like "aabaabaabaab ..." (repeated several hundred times) may compress more slowly than normal. Versions 0.9.5 and above fare much better than previous versions in this respect. The ratio between worst-case and average-case compression time is in the region of 10:1. For previous versions, this figure was more like 100:1. You can use the -vvvv option to monitor progress in great detail, if you want.

Decompression speed is unaffected by these phenomena.

*bzip2* usually allocates several megabytes of memory to operate in, and then charges all over it in a fairly random fashion. This means that performance, both for compressing and decompressing, is largely determined by the speed at which your machine can service cache misses. Because of this, small changes to the code to reduce the miss rate have been observed to give disproportionately large performance improvements. I imagine *bzip2* will perform best on machines with very large caches.

## CAVEATS

I/O error messages are not as helpful as they could be. *bzip2* tries hard to detect I/O errors and exit cleanly, but the details of what the problem is sometimes seem rather misleading.

This manual page pertains to version 1.0.3 of *bzip2*. Compressed data created by this version is entirely forwards and backwards compatible with the previous public releases, versions 0.1pl2, 0.9.0, 0.9.5, 1.0.0, 1.0.1 and 1.0.2, but with the following exception: 0.9.0 and above can correctly decompress multiple concatenated compressed files. 0.1pl2 cannot do this; it will stop after decompressing just the first file in the stream.

*bzip2recover* versions prior to 1.0.2 used 32-bit integers to represent bit positions in compressed files, so they could not handle compressed files more than 512 megabytes long. Versions 1.0.2 and above use 64-bit ints on some platforms which support them (GNU supported targets, and Windows). To establish whether or not *bzip2recover* was built with such a limitation, run it without arguments. In any event you can build yourself an unlimited version if you can recompile it with `MaybeUInt64` set to be an unsigned 64-bit integer.

## AUTHOR

Julian Seward, [jsewardbzip.org](http://jsewardbzip.org).

<http://www.bzip.org>

The ideas embodied in *bzip2* are due to (at least) the following people: Michael Burrows and David Wheeler (for the block sorting transformation), David Wheeler (again, for the Huffman coder), Peter Fenwick (for the structured coding model in the original *bzip*, and many refinements), and Alistair Moffat, Radford Neal and Ian Witten (for the arithmetic coder in the original *bzip*). I am much indebted for their help, support and advice. See the manual in the source distribution for pointers to sources of documentation. Christian von Roques encouraged me to look for faster sorting algorithms, so as to speed up compression. Bela Lubkin encouraged me to improve the worst-case compression performance. Donna Robinson XMLised the documentation. The *bz\** scripts are derived from those of GNU *gzip*. Many people sent patches, helped with portability problems, lent machines, gave advice and were generally helpful.

**NAME**

**bzip2**, **bunzip2** – a block-sorting file compressor, v1.0.3  
**bzcat** – decompresses files to stdout  
**bzip2recover** – recovers data from damaged bzip2 files

**SYNOPSIS**

**bzip2** [ **-cdfkqstvwVL123456789** ] [ *filenames ...* ]  
**bunzip2** [ **-fkvsVL** ] [ *filenames ...* ]  
**bzcat** [ **-s** ] [ *filenames ...* ]  
**bzip2recover** *filename*

**DESCRIPTION**

*bzip2* compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding. Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors, and approaches the performance of the PPM family of statistical compressors.

The command-line options are deliberately very similar to those of *GNU gzip*, but they are not identical.

*bzip2* expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the name "original\_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

*bzip2* and *bunzip2* will by default not overwrite existing files. If you want this to happen, specify the **-f** flag.

If no file names are specified, *bzip2* compresses from standard input to standard output. In this case, *bzip2* will decline to write compressed output to a terminal, as this would be entirely incomprehensible and therefore pointless.

*bunzip2* (or *bzip2 -d*) decompresses all specified files. Files which were not created by *bzip2* will be detected and ignored, and a warning issued. *bzip2* attempts to guess the filename for the decompressed file from that of the compressed file as follows:

```
filename.bz2 becomes filename
filename.bz becomes filename
filename.tbz2 becomes filename.tar
filename.tbz becomes filename.tar
anyothername becomes anyothername.out
```

If the file does not end in one of the recognised endings, *.bz2*, *.bz*, *.tbz2* or *.tbz*, *bzip2* complains that it cannot guess the name of the original file, and uses the original name with *.out* appended.

As with compression, supplying no filenames causes decompression from standard input to standard output.

*bunzip2* will correctly decompress a file which is the concatenation of two or more compressed files. The result is the concatenation of the corresponding uncompressed files. Integrity testing (**-t**) of concatenated compressed files is also supported.

You can also compress or decompress files to the standard output by giving the **-c** flag. Multiple files may be compressed and decompressed like this. The resulting outputs are fed sequentially to stdout.

Compression of multiple files in this manner generates a stream containing multiple compressed file representations. Such a stream can be decompressed correctly only by *bzip2* version 0.9.0 or later. Earlier versions of *bzip2* will stop after decompressing the first file in the stream.

*bzcat* (or *bzip2 -dc*) decompresses all specified files to the standard output.

*bzip2* will read arguments from the environment variables *BZIP2* and *BZIP*, in that order, and will process them before any arguments read from the command line. This gives a convenient way to supply default arguments.

Compression is always performed, even if the compressed file is slightly larger than the original. Files of less than about one hundred bytes tend to get larger, since the compression mechanism has a constant overhead in the region of 50 bytes. Random data (including the output of most file compressors) is coded at about 8.05 bits per byte, giving an expansion of around 0.5%.

As a self-check for your protection, *bzip2* uses 32-bit CRCs to make sure that the decompressed version of a file is identical to the original. This guards against corruption of the compressed data, and against undetected bugs in *bzip2* (hopefully very unlikely). The chances of data corruption going undetected is microscopic, about one chance in four billion for each file processed. Be aware, though, that the check occurs upon decompression, so it can only tell you that something is wrong. It can't help you recover the original uncompressed data. You can use *bzip2recover* to try to recover data from damaged files.

Return values: 0 for a normal exit, 1 for environmental problems (file not found, invalid flags, I/O errors, &c), 2 to indicate a corrupt compressed file, 3 for an internal consistency error (eg, bug) which caused *bzip2* to panic.

## OPTIONS

### **-c --stdout**

Compress or decompress to standard output.

### **-d --decompress**

Force decompression. *bzip2*, *bunzip2* and *bzcat* are really the same program, and the decision about what actions to take is done on the basis of which name is used. This flag overrides that mechanism, and forces *bzip2* to decompress.

### **-z --compress**

The complement to *-d*: forces compression, regardless of the invocation name.

### **-t --test**

Check integrity of the specified file(s), but don't decompress them. This really performs a trial decompression and throws away the result.

### **-f --force**

Force overwrite of output files. Normally, *bzip2* will not overwrite existing output files. Also forces *bzip2* to break hard links to files, which it otherwise wouldn't do.

*bzip2* normally declines to decompress files which don't have the correct magic header bytes. If forced (*-f*), however, it will pass such files through unmodified. This is how GNU *gzip* behaves.

### **-k --keep**

Keep (don't delete) input files during compression or decompression.

### **-s --small**

Reduce memory usage, for compression, decompression and testing. Files are decompressed and tested using a modified algorithm which only requires 2.5 bytes per block byte. This means any file can be decompressed in 2300k of memory, albeit at about half the normal speed.

During compression, *-s* selects a block size of 200k, which limits memory use to around the same

figure, at the expense of your compression ratio. In short, if your machine is low on memory (8 megabytes or less), use `-s` for everything. See MEMORY MANAGEMENT below.

**-q --quiet**

Suppress non-essential warning messages. Messages pertaining to I/O errors and other critical events will not be suppressed.

**-v --verbose**

Verbose mode -- show the compression ratio for each file processed. Further `-v`'s increase the verbosity level, spewing out lots of information which is primarily of interest for diagnostic purposes.

**-L --license -V --version**

Display the software version, license terms and conditions.

**-1 (or --fast) to -9 (or --best)**

Set the block size to 100 k, 200 k .. 900 k when compressing. Has no effect when decompressing. See MEMORY MANAGEMENT below. The `--fast` and `--best` aliases are primarily for GNU gzip compatibility. In particular, `--fast` doesn't make things significantly faster. And `--best` merely selects the default behaviour.

**--**

Treats all subsequent arguments as file names, even if they start with a dash. This is so you can handle files with names beginning with a dash, for example: `bzip2 -- -myfilename`.

**--repetitive-fast --repetitive-best**

These flags are redundant in versions 0.9.5 and above. They provided some coarse control over the behaviour of the sorting algorithm in earlier versions, which was sometimes useful. 0.9.5 and above have an improved algorithm which renders these flags irrelevant.

## MEMORY MANAGEMENT

*bzip2* compresses large files in blocks. The block size affects both the compression ratio achieved, and the amount of memory needed for compression and decompression. The flags `-1` through `-9` specify the block size to be 100,000 bytes through 900,000 bytes (the default) respectively. At decompression time, the block size used for compression is read from the header of the compressed file, and *bunzip2* then allocates itself just enough memory to decompress the file. Since block sizes are stored in compressed files, it follows that the flags `-1` to `-9` are irrelevant to and so ignored during decompression.

Compression and decompression requirements, in bytes, can be estimated as:

Compression:  $400k + (8 \times \text{block size})$

Decompression:  $100k + (4 \times \text{block size})$ , or  
 $100k + (2.5 \times \text{block size})$

Larger block sizes give rapidly diminishing marginal returns. Most of the compression comes from the first two or three hundred k of block size, a fact worth bearing in mind when using *bzip2* on small machines. It is also important to appreciate that the decompression memory requirement is set at compression time by the choice of block size.

For files compressed with the default 900k block size, *bunzip2* will require about 3700 kbytes to decompress. To support decompression of any file on a 4 megabyte machine, *bunzip2* has an option to decompress using approximately half this amount of memory, about 2300 kbytes. Decompression speed is also halved, so you should use this option only where necessary. The relevant flag is `-s`.

In general, try and use the largest block size memory constraints allow, since that maximises the compression achieved. Compression and decompression speed are virtually unaffected by block size.

Another significant point applies to files which fit in a single block -- that means most files you'd encounter using a large block size. The amount of real memory touched is proportional to the size of the file, since

the file is smaller than a block. For example, compressing a file 20,000 bytes long with the flag -9 will cause the compressor to allocate around 7600k of memory, but only touch 400k + 20000 \* 8 = 560 kbytes of it. Similarly, the decompressor will allocate 3700k but only touch 100k + 20000 \* 4 = 180 kbytes.

Here is a table which summarises the maximum memory usage for different block sizes. Also recorded is the total compressed size for 14 files of the Calgary Text Compression Corpus totalling 3,141,622 bytes. This column gives some feel for how compression varies with block size. These figures tend to understate the advantage of larger block sizes for larger files, since the Corpus is dominated by smaller files.

| Flag | Compress<br>usage | Decompress<br>usage | Decompress<br>-s usage | Corpus<br>Size |
|------|-------------------|---------------------|------------------------|----------------|
| -1   | 1200k             | 500k                | 350k                   | 914704         |
| -2   | 2000k             | 900k                | 600k                   | 877703         |
| -3   | 2800k             | 1300k               | 850k                   | 860338         |
| -4   | 3600k             | 1700k               | 1100k                  | 846899         |
| -5   | 4400k             | 2100k               | 1350k                  | 845160         |
| -6   | 5200k             | 2500k               | 1600k                  | 838626         |
| -7   | 6100k             | 2900k               | 1850k                  | 834096         |
| -8   | 6800k             | 3300k               | 2100k                  | 828642         |
| -9   | 7600k             | 3700k               | 2350k                  | 828642         |

## RECOVERING DATA FROM DAMAGED FILES

*bzip2* compresses files in blocks, usually 900kbytes long. Each block is handled independently. If a media or transmission error causes a multi-block .bz2 file to become damaged, it may be possible to recover data from the undamaged blocks in the file.

The compressed representation of each block is delimited by a 48-bit pattern, which makes it possible to find the block boundaries with reasonable certainty. Each block also carries its own 32-bit CRC, so damaged blocks can be distinguished from undamaged ones.

*bzip2recover* is a simple program whose purpose is to search for blocks in .bz2 files, and write each block out into its own .bz2 file. You can then use *bzip2* -t to test the integrity of the resulting files, and decompress those which are undamaged.

*bzip2recover* takes a single argument, the name of the damaged file, and writes a number of files "rec00001file.bz2", "rec00002file.bz2", etc, containing the extracted blocks. The output filenames are designed so that the use of wildcards in subsequent processing -- for example, "bzip2 -dc rec\*file.bz2 > recovered\_data" -- processes the files in the correct order.

*bzip2recover* should be of most use dealing with large .bz2 files, as these will contain many blocks. It is clearly futile to use it on damaged single-block files, since a damaged block cannot be recovered. If you wish to minimise any potential data loss through media or transmission errors, you might consider compressing with a smaller block size.

## PERFORMANCE NOTES

The sorting phase of compression gathers together similar strings in the file. Because of this, files containing very long runs of repeated symbols, like "aabaabaabaab ..." (repeated several hundred times) may compress more slowly than normal. Versions 0.9.5 and above fare much better than previous versions in this respect. The ratio between worst-case and average-case compression time is in the region of 10:1. For previous versions, this figure was more like 100:1. You can use the -vvvv option to monitor progress in great detail, if you want.

Decompression speed is unaffected by these phenomena.



*bzip2* usually allocates several megabytes of memory to operate in, and then charges all over it in a fairly random fashion. This means that performance, both for compressing and decompressing, is largely determined by the speed at which your machine can service cache misses. Because of this, small changes to the code to reduce the miss rate have been observed to give disproportionately large performance improvements. I imagine *bzip2* will perform best on machines with very large caches.

## CAVEATS

I/O error messages are not as helpful as they could be. *bzip2* tries hard to detect I/O errors and exit cleanly, but the details of what the problem is sometimes seem rather misleading.

This manual page pertains to version 1.0.3 of *bzip2*. Compressed data created by this version is entirely forwards and backwards compatible with the previous public releases, versions 0.1pl2, 0.9.0, 0.9.5, 1.0.0, 1.0.1 and 1.0.2, but with the following exception: 0.9.0 and above can correctly decompress multiple concatenated compressed files. 0.1pl2 cannot do this; it will stop after decompressing just the first file in the stream.

*bzip2recover* versions prior to 1.0.2 used 32-bit integers to represent bit positions in compressed files, so they could not handle compressed files more than 512 megabytes long. Versions 1.0.2 and above use 64-bit ints on some platforms which support them (GNU supported targets, and Windows). To establish whether or not *bzip2recover* was built with such a limitation, run it without arguments. In any event you can build yourself an unlimited version if you can recompile it with `MaybeUInt64` set to be an unsigned 64-bit integer.

## AUTHOR

Julian Seward, [jsewardbzip.org](http://jsewardbzip.org).

<http://www.bzip.org>

The ideas embodied in *bzip2* are due to (at least) the following people: Michael Burrows and David Wheeler (for the block sorting transformation), David Wheeler (again, for the Huffman coder), Peter Fenwick (for the structured coding model in the original *bzip*, and many refinements), and Alistair Moffat, Radford Neal and Ian Witten (for the arithmetic coder in the original *bzip*). I am much indebted for their help, support and advice. See the manual in the source distribution for pointers to sources of documentation. Christian von Roques encouraged me to look for faster sorting algorithms, so as to speed up compression. Bela Lubkin encouraged me to improve the worst-case compression performance. Donna Robinson XMLised the documentation. The *bz\** scripts are derived from those of GNU *gzip*. Many people sent patches, helped with portability problems, lent machines, gave advice and were generally helpful.

**NAME**

*bzmore*, *bzless* – file perusal filter for crt viewing of bzip2 compressed text

**SYNOPSIS**

**bzmore** [ name ... ]

**bzless** [ name ... ]

**NOTE**

In the following description, *bzless* and *less* can be used interchangeably with *bzmore* and *more*.

**DESCRIPTION**

*Bzmore* is a filter which allows examination of compressed or plain text files one screenful at a time on a soft-copy terminal. *bzmore* works on files compressed with *bzip2* and also on uncompressed files. If a file does not exist, *bzmore* looks for a file of the same name with the addition of a .bz2 suffix.

*Bzmore* normally pauses after each screenful, printing --More-- at the bottom of the screen. If the user then types a carriage return, one more line is displayed. If the user hits a space, another screenful is displayed. Other possibilities are enumerated later.

*Bzmore* looks in the file */etc/termcap* to determine terminal characteristics, and to determine the default window size. On a terminal capable of displaying 24 lines, the default window size is 22 lines. Other sequences which may be typed when *bzmore* pauses, and their effects, are as follows (*i* is an optional integer argument, defaulting to 1) :

- i* <space> display *i* more lines, (or another screenful if no argument is given)
- ^D display 11 more lines (a “scroll”). If *i* is given, then the scroll size is set to *i*.
- d same as ^D (control-D)
- i* z same as typing a space except that *i*, if present, becomes the new window size. Note that the window size reverts back to the default at the end of the current file.
- i* s skip *i* lines and print a screenful of lines
- i* f skip *i* screenfuls and print a screenful of lines
- q or Q quit reading the current file; go on to the next (if any)
- e or q When the prompt --More--(Next file: *file*) is printed, this command causes *bzmore* to exit.
- s When the prompt --More--(Next file: *file*) is printed, this command causes *bzmore* to skip the next file and continue.
- = Display the current line number.
- i* /*expr* search for the *i*-th occurrence of the regular expression *expr*. If the pattern is not found, *bzmore* goes on to the next file (if any). Otherwise, a screenful is displayed, starting two lines before the place where the expression was found. The user's erase and kill characters may be used to edit the regular expression. Erasing back past the first column cancels the search command.
- i* n search for the *i*-th occurrence of the last regular expression entered.
- !*command* invoke a shell with *command*. The character '!' in "*command*" are replaced with the previous shell command. The sequence "\!" is replaced by "!".
- :q or :Q quit reading the current file; go on to the next (if any) (same as q or Q).
- . (dot) repeat the previous command.

The commands take effect immediately, i.e., it is not necessary to type a carriage return. Up to the time when the command character itself is given, the user may hit the line kill character to cancel the numerical argument being formed. In addition, the user may hit the erase character to redisplay the --More-- message.

At any time when output is being sent to the terminal, the user can hit the quit key (normally control-\\).

*Bzmore* will stop sending output, and will display the usual --More-- prompt. The user may then enter one of the above commands in the normal manner. Unfortunately, some output is lost when this is done, due to the fact that any characters waiting in the terminal's output queue are flushed when the quit signal occurs.

The terminal is set to *noecho* mode by this program so that the output can be continuous. What you type will thus not show on your terminal, except for the / and ! commands.

If the standard output is not a teletype, then *bzmore* acts just like *bzcat*, except that a header is printed before each file.

**FILES**

/etc/termcap                      Terminal data base

**SEE ALSO**

more(1), less(1), bzip2(1), bzdiff(1), bzgrep(1)

**NAME**

catcomp – cat a file to stdout out of a compressed archive file

**SYNOPSIS**

**edrc/bin/catcomp** [ **-h** ]

**catcomp** *file path*

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

With **catcomp** you cat a file out of a compressed archive without decompressing the whole archive file. To list the contents of an archive file use the **lscomp** or **llcomp** command.

**OPTIONS**

**-h**           usage message

*file*           filename of the compressed archive file. This file is a file with the suffixes **.tgz**, **.tar.gz**, **.tar.bz**, **.tar.bz2**, **.tar.Z**, **.tar.zip**, **.cpio**, **.cpio.gz**, **.cpio.bz**, **.cpio.bz2**, **.cpio.xz**, **.cpio.Z**, **.cpio.zip**, **.depot.gz**, **.depot**, **.epub**, **.zip**, **.jar**, **.whl**, **.rock**, **.rpm**, **.deb** or **.bw**.

*path*           complete path and filename of the file inside the archive you like to display.

**ENVIRONMENT**

The following environment variables influence the behavior of the **catcomp** command:

**\$TMPDIR**

if you like to display files that are bigger then the free disk space available in **/tmp** this variable has to be set to a directory with sufficient disk space available. You have to ensure that the directory you specify exists and is writeable by the user invoking the **catcomp** command.

**EXIT STATUS**

**1**           specified archive file not found.

- 2** file format of specified archive file not recognized. The file format is determined by the file suffix only.
- 4** usage displayed.
- 5** file not found in archive. If this error occurs, try also **lscomp** *filename* to figure out if the archive is probably corrupt.
- 6** error while decompressing a compressed archive. This error occurs if the file is not compressed with the command that is normally assigned with the file suffix.  
For instance: the file **my\_archive.tar.gz** was not really compressed with **gzip** or the file is corrupt.

## SEE ALSO

**edrcintro(1)**, **lscomp(1)**, **llcomp(1)**, **ar(1)**, **bzip2(1)**, **bwcreate(1)**, **bwcompare(1)**, **zip(1)**, **unzip(1)**, **compress(1)**, **cpio(1)**, **dpkg(1)**, **rpm(1)**, **jar(1)**, **tar(1)**, **gzip(1)**, **xz(1)**

## BUGS

-

## AUTHOR

catcomp was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

## COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

**ccrypt** – encrypt and decrypt files and streams

**SYNOPSIS**

```
ccrypt [mode] [options] [file...]
ccencrypt [options] [file...]
ccdecrypt [options] [file...]
ccat [options] file...
```

**DESCRIPTION**

**ccrypt** is a utility for encrypting and decrypting files and streams. It was designed to replace the standard unix **crypt** utility, which is notorious for using a very weak encryption algorithm. **ccrypt** is based on the Rijndael block cipher, which was also chosen by the U.S. government as the Advanced Encryption Standard (AES, see <http://www.nist.gov/aes>). This cipher is believed to provide very strong cryptographic security.

Unlike unix **crypt**, the algorithm provided by **ccrypt** is not symmetric, i.e., one must specify whether to encrypt or decrypt. The most common way to invoke **ccrypt** is via the commands **ccencrypt** and **ccdecrypt**.

Encryption and decryption depends on a keyword (or key phrase) supplied by the user. By default, the user is prompted to enter a keyword from the terminal. Keywords can consist of any number of characters, and all characters are significant (although **ccrypt** internally hashes the key to 256 bits). Longer keywords provide better security than short ones, since they are less likely to be discovered by exhaustive search.

**MODES**

**ccrypt** can operate in five different modes. If more than one mode is specified, the last one specified takes precedence. The aliases **ccencrypt**, **ccdecrypt**, and **ccat** are provided as a convenience; they are equivalent to **ccrypt -e**, **ccrypt -d**, and **ccrypt -c**, respectively.

- e, --encrypt**      Encrypt. This is the default mode. If filename arguments are given, encrypt the files and append the suffix **.cpt** to their names. Otherwise, run as a filter.
- d, --decrypt**      Decrypt. If filename arguments are given, decrypt the files and strip the suffix **.cpt** from the filenames, if present. Otherwise, run as a filter.
- c, --cat**          Decrypt one or more files to standard output. If no filename arguments are given, decrypt as a filter. Implies **-l**.
- x, --keychange**    Change the key of encrypted data. In this mode, **ccrypt** prompts for two passwords: the old one and the new one. If filename arguments are given, modify the files. Otherwise, run as a filter.
- u, --unixcrypt**    Simulate the old unix crypt command. Note: the cipher used by unix crypt has been broken and is not secure. Please use this option only to decrypt existing files. If filename arguments are given, decrypt the files to stdout. Otherwise, run as a filter. Note that for the unix crypt format, there is no easy way to detect whether a given key matches or not; thus, for safety, this mode does not overwrite files.

**OPTIONS**

The following options are supported in addition to the modes described above:

- b, --brave**        When reading an encryption key from the terminal, ask the user only once for the key. By default, **ccrypt** will ask the user to enter such keys twice, as a safeguard against accidentally destroying data due to a mistyped key. Using the **--brave** option disables this safety check. Never use it, unless you know what you are doing. See also **--timid**.
- E var, --envvar var**    Read the keyword from the environment variable *var*. Note that this might be unsafe on certain systems, namely where users can use the **ps** command to see the environment of a process owner by another user. On most modern systems, however, such behavior of **ps**

- is disabled and the **-E** option should be safe there. Also, as an added security measure, **ccrypt** erases the keyword from its environment immediately after reading it.
- f, --force** Overwrite existing files or operate on write-protected files without asking any questions. Also, override **ccrypt**'s reluctance to write or read encrypted data to or from a terminal.
  - F var, --envvar2 var** Same as **-E**, except for second keyword (in keychange mode).
  - h, --help** Help. Print usage information and exit.
  - H key, --key2 key** Same as **-K**, except for second keyword (in keychange mode).
  - k file, --keyfile file** Read the keyword as the first line from the named file. In keychange mode, two keywords are read as the first two lines of the file. The filename "-" may be given for reading keywords from standard input. Using the **-k** option and sending the keyword on stdin is probably the safest way to pass a keyword to **ccrypt** from another program or script.
  - K key, --key key** Specify the keyword on the command line. This is unsafe, because any other user can see the command line by running the **ps** command. Only use this option for testing purposes, and never with a real keyword.
  - l, --symlinks** Force encryption/decryption of symbolic links. By default, symbolic links are ignored except in cat or unixcrypt mode. Note that with the **-l** option, encryption/decryption of a symbolic link causes the suffix **.cpt** to be added/removed from the name of the link, not the name of the file pointed to.
  - L, --license** Print license info and exit.
  - m, --mismatch** Normally, **ccrypt** refuses to decrypt data with a key that does not seem to match. The **-m** option overrides this restriction. This can sometimes be useful in recovering data from a corrupted file (see RECOVERING DATA FROM CORRUPTED FILES). To avoid irretrievable loss of data when decrypting with a wrong key, this option cannot be used with modes that overwrite the input file.
  - P prompt, --prompt prompt** Use *prompt* instead of the default prompt "Enter encryption/decryption key: ". This may be useful in some shell scripts.
  - q, --quiet** Suppress most warnings.
  - Q prompt, --prompt2 prompt** Same as **-P**, except for second keyword (in keychange mode).
  - r, --recursive** Traverse subdirectories recursively.
  - R, --rec-symlinks** Traverse subdirectories recursively, and also follow symbolic links to subdirectories.
  - s, --strictsuffix** Refuse to encrypt files which already have the **.cpt** suffix (or that selected with **-S**). This can be useful when adding some files to a directory of already encrypted files. This option has no effect in decryption or keychange mode.
  - S .suf, --suffix .suf** Use the suffix *.suf* instead of the default suffix **.cpt**.
  - t, --timid** When reading an encryption key from the terminal, ask the user to enter the key twice. If the two entered keys are not identical, abort. This is a safeguard against accidentally destroying data by encrypting it with a mistyped key. Note: this behavior is now the default, and can be overridden with the **--brave** option.
  - T, --tmpfiles** This option causes **ccrypt** to use temporary files during encryption/decryption, rather than overwriting the file contents destructively. This method leaves the original file

contents lying around in unused sectors of the file system, and thus is less secure than the default behavior. However, in situations where this loss of security is not important, the **--tmpfiles** option can provide a measure of protection against data being corrupted due to a system crash in the middle of overwriting a file.

- v, --verbose** Print progress information to stderr.
- V, --version** Print version info and exit.
- End of options. Any remaining arguments are interpreted as filenames. This also turns off filter mode, even if zero filenames follow. This might be useful in the context of shell pattern expansion; **ccrypt -- \*** will behave correctly even if no files match the pattern **\***.

## NOTES ON USAGE

The user interface of **ccrypt** intentionally resembles that of GNU **gzip**, although it is not identical. When invoked with filename arguments, **ccrypt** normally modifies the files in place, overwriting their old content. Unlike **gzip**, the output is not first written to a temporary file; instead, the data is literally overwritten. For encryption, this is usually the desired behavior, since one does not want copies of the unencrypted data to remain in hidden places in the file system. The disadvantage is that if **ccrypt** is interrupted in the middle of writing to a file, the file will end up in a corrupted, partially encrypted state. However, in such cases it is possible to recover most of the data; see RECOVERING DATA FROM CORRUPTED FILES below. If you want to force **ccrypt** to use temporary files, use the **--tmpfiles** option.

When **ccrypt** receives an interrupt signal (Ctrl-C) while updating a file in place, it does not exit immediately, but rather delays the exit until after it finishes writing to the current file. This is to prevent files from being partially overwritten and thus corrupted. If you want to force **ccrypt** to exit immediately, just press Ctrl-C twice quickly.

The encryption algorithm used by **ccrypt** uses a random seed which is different each time. As a result, encrypting the same file twice will never yield the same result. The advantage of this method is that similarities in plaintext do not lead to similarities in ciphertext; there is no way of telling whether the content of two encrypted files is similar or not.

Because of the use of a random seed, decrypting and re-encrypting a file with the same key will not lead to an identical file. It is primarily for this reason that **ccrypt** refuses to decrypt files with a non-matching key; if this were allowed, there would be no way afterwards to restore the original file, and the data would be irretrievably lost.

When overwriting files, special care is taken with hard links and symbolic links. Each physical file (i.e., each inode) is processed at most once, no matter how many paths to it are encountered on the command line or in subdirectories traversed recursively. For each file which has multiple hard links, a warning is printed, to alert the user that not all paths to the file might have been properly renamed. Symbolic links are ignored except in cat mode, or unless the **-I** or **-R** option is given.

Unlike **gzip**, **ccrypt** does not complain about files that have improper suffixes. It is legal to doubly encrypt a file. It is also legal to decrypt a file which does not have the **.cpt** suffix, provided the file contains valid data for the given decryption key. Use the **--strictsuffix** option if you want to prevent **ccrypt** from encrypting files which already have a **.cpt** suffix.

Regarding encryption and compression: encrypted data is statistically indistinguishable from random data, and thus it cannot be compressed. But of course it is possible to compress the data first, then encrypt it. Suggested file suffixes are **.gz.cpt** or **.gzc**.

## RECOVERING DATA FROM CORRUPTED FILES

Encrypted data might be corrupted for a number of reasons. For instance, a file might have been partially encrypted or decrypted if **ccrypt** was interrupted while processing the file. Or data might be corrupted by a software or hardware error, or during transmission over a network. The encryption algorithm used by



**ccrypt** is designed to allow recovery from errors. In general, only a few bytes of data will be lost near where the error occurred.

Data encrypted by **ccrypt** can be thought of as a sequence of 32-byte blocks. To decrypt a particular block, **ccrypt** only needs to know the decryption key, the data of the block itself, and the data of the block immediately preceding it. **ccrypt** cannot tell whether a block is corrupted or not, except the very first block, which is special. Thus, if the encrypted data has been altered in the middle or near the end of a file, **ccrypt** can be run to decrypt it as usual, and most of the data will be decrypted correctly, except near where the corruption occurred.

The very first block of encrypted data is special, because it does not actually correspond to any plaintext data; this block holds the random seed generated at encryption time. **ccrypt** also uses the very first block to decide whether the given keyword matches the data or not. If the first block has been corrupted, **ccrypt** will likely decide that the keyword does not match; in such cases, the **-m** option can be used to force **ccrypt** to decrypt the data anyway.

If a file contains some encrypted and some unencrypted data, or data encrypted with two different keys, one should decrypt the entire file with each applicable key, and then piece together the meaningful parts manually.

Finally, decryption will only produce meaningful results if the data is aligned correctly along block boundaries. If the block boundary information has been lost, one has to try all 32 possibilities.

## DESCRIPTION OF THE CIPHER

*Block ciphers* operate on data segments of a fixed length. For instance, the Rijndael block cipher used in **ccrypt** has a block length of 32 bytes or 256 bits. Thus, this cipher encrypts 32 bytes at a time.

*Stream ciphers* operate on data streams of any length. There are several standard modes for operating a block cipher as a stream cipher. One such standard is *Cipher Feedback (CFB)*, defined in FIPS 81 and ANSI X3.106-1983. **ccrypt** implements a stream cipher by operating the Rijndael block cipher in CFB mode.

Let  $P[i]$  and  $C[i]$  be the  $i$ th block of the plaintext and ciphertext, respectively. CFB mode specifies that

$$C[i] = P[i] \oplus E(k, C[i-1])$$

Here  $\oplus$  denotes the bitwise exclusive or function, and  $E(k, x)$  denotes the encryption of the block  $x$  under the key  $k$  using the block cipher. Thus, each block of the ciphertext is calculated from the corresponding block of plaintext and the previous block of ciphertext. Note that in fact, each byte of  $P[i]$  can be calculated from the corresponding byte of  $C[i]$ , so that the stream cipher can be applied to one byte at a time. In particular, the stream length need not be a multiple of the block size.

Assuming that blocks are numbered starting from 0, a special "initial" ciphertext block  $C[-1]$  is needed to provide the base case for the above formula. This value  $C[-1]$  is called the *initialization vector* or *seed*. The seed is chosen at encryption time and written as the first block of the encrypted stream. It is important never to use the same seed more than once; otherwise, the two resulting ciphertext blocks  $C[0]$  could be related by a simple xor to obtain information about the corresponding plaintext blocks  $P[0]$ . If the same seed is never reused, CFB is provably as secure as the underlying block cipher.

In **ccrypt**, the seed is constructed as follows: first, a combination of the host name, current time, process id, and an internal counter are hashed into a 28-byte value, using a cryptographic hash function. A fixed four-byte "magic number" is combined with this value, and the resulting 32-byte value is encrypted by one round of the Rijndael block cipher with the given key. The result is used as the seed and appended to the beginning of the ciphertext. The use of the magic number allows **ccrypt** to detect non-matching keys before decryption.

## SECURITY

**ccrypt** is believed to provide very strong cryptographic security, equivalent to that of the Rijndael cipher with 256-bit block size and 256-bit key size. Another version of the Rijndael cipher (with a smaller block size) is used in the U.S. government's Advanced Encryption Standard (AES, see <http://www.nist.gov/aes>). Therefore, this cipher is very well studied and subject to intensive public scrutiny. This scrutiny has a positive effect on the cipher's security. In particular, if an exploitable weakness in this cipher were ever discovered, this would become widely publicized.

In practical terms, the security of **ccrypt** means that, without knowledge of the encryption key, it is effectively impossible to obtain any information about the plaintext from a given ciphertext. This is true even if a large number of plaintext-ciphertext pairs are already known for the same key. Moreover, because **ccrypt** uses a key size of 256 bits, an exhaustive search of the key space is not feasible, at least as long as sufficiently long keys are actually used in practice. No cipher is secure if users choose insecure keywords.

On the other hand, **ccrypt** does not attempt to provide *data integrity*, i.e., it will not attempt to detect whether the ciphertext was modified after encryption. In particular, encrypted data can be truncated, leaving the corresponding decrypted data also truncated, but otherwise consistent. If one needs to ensure data integrity as well as secrecy, this can be achieved by other methods. The recommended method is to prepend a cryptographic hash (for instance, an MD5 or SHA-1 hash) to the data before encryption.

**ccrypt** does not claim to provide any particular safeguards against information leaking via the local operating system. While reasonable precautions are taken, there is no guarantee that keywords and plaintexts have been physically erased after encryption is completed; parts of such data might still exist in memory or on disk. **ccrypt** does not currently use privileged memory pages.

When encrypting files, **ccrypt** by default accesses them in read-write mode. This normally causes the original file to be physically overwritten, but on some file systems, this might not be the case.

Note that the use of the **-K** option is unsafe in a multiuser environment, because the command line of a process is visible to other users running the **ps** command. The use of the **-E** option is potentially unsafe for the same reason, although recent versions of **ps** don't tend to display environment information to other users. The use of the **-T** option is unsafe for encryption because the original plaintext will remain in unused sectors of the file system.

## EMACS PACKAGE

There is an emacs package for reading and writing encrypted files. (Note that this package currently only works with emacs, not with xemacs.) This package hooks into the low-level file I/O functions of emacs, prompting the user for a password where appropriate. It is implemented in much the same way as support for compressed files; in fact, the package, whose name is `jka-compr-ccrypt`, is based directly on the `jka-compr` package which is part of GNU Emacs. It handles both encrypted and compressed files (although currently not encrypted compressed files).

To use the package, simply load `jka-compr-ccrypt`, then edit as usual. When you open a file with the `".cpt"` extension, emacs will prompt you for a password for the file. It will remember the password for the buffer, and when you save the file later, it will be automatically encrypted again (provided you save it with a `".cpt"` extension). Except for the password prompt, the operation of the package should be transparent to the user. The package also handles compressed `".gz"`, `".bz2"`, and `".Z"` files, and it should be used instead of, not in addition to, `jka-compr`. The command `M-x ccrypt-set-buffer-password` can be used to change the current password of a buffer.

The simplest way to use this package is to include the lines

```
(setq load-path (cons "path" load-path))
(require 'jka-compr-ccrypt "jka-compr-ccrypt.el")
```

in your `.emacs` file, where *path* is the directory which holds the file `jka-compr-ccrypt.el`. You may also need

to disable loading of the old jka-compr package, since the two packages are not compatible with each other (and in any case, jka-compr-ccrypt contains all the functionality of jka-compr).

Limitations of the emacs package: there is no guarantee that unencrypted information cannot leak to the file system; in fact, the package sometimes writes unencrypted data to temporary files. However, auto-saved files are normally treated correctly (i.e., encrypted). For details, see the comments in the file jka-compr-ccrypt.el.

## EXIT STATUS

The exit status is 0 on successful completion, and non-zero otherwise. An exit status of 1 means illegal command line, 2 is out of memory, 3 is a fatal i/o error, 4 is a non-matching key or wrong file format, 6 is interrupt, 7 is mistyped key in **--timid** mode, and 8 is a non-fatal i/o error.

Fatal i/o errors are those which occur while processing a file which is already open. Such errors cause ccrypt to abort its operation immediately with an exit status of 3. Non-fatal i/o errors are those which occur while handling files which are not already open; typically, such errors are caused by files that are missing, not readable, or can't be created. When encountering a non-fatal i/o error, ccrypt simply continues to process the next available input file. The exit status of 8 is delayed until after all the files have been processed.

Non-matching keys and wrong file formats are also considered non-fatal errors, and cause ccrypt to continue with processing the next available input file. In this case, an exit status of 4 is given after all the files have been processed. If there is a conflict between exit status 4 and 8, then 8 is returned.

The former exit status 5 ("wrong file format") has been eliminated, and is now covered under exit status 4 ("non-matching key or wrong file format"). Note that ccrypt does not really have a "file format" in the proper sense of the word; any file of length at least 32 bytes is potentially a valid encrypted file.

## BUGS

While **ccrypt** can handle keywords of arbitrary length, some operating systems limit the length of an input line to 1024 characters.

The renaming of files (adding or removing the **.cpt** suffix) can go wrong if a filename is repeated on the command line. In this case, the file is only encrypted/decrypted once, but the suffix may be added or removed several times. This is because **ccrypt** thinks it encountered different hardlinks for the same file.

The **--strictsuffix** option can behave in unexpected ways if one file has several hardlinks of which some have the suffix and some don't. In this case, the inode will be encrypted/decrypted, but the suffix will be changed only for those filenames that allow it. Similarly, if a file cannot be renamed because a file of the given name already exists, the file may still be encrypted/decrypted if it has another hardlink.

## VERSION

1.7

## AUTHOR

Peter Selinger <selinger at users.sourceforge.net>

## COPYRIGHT

Copyright (C) 2000-2004 Peter Selinger

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA. See also <http://www.gnu.org/>.

cfg2html is a little utility program that can be used to collect the necessary system configuration files and system set-up information to both an ASCII and an HTML file. Simple to use and very helpful when preparing for disaster recovery situations.

## NAME

cfg2html – creates nice HTML and plain ASCII documentation of your GNU/Linux systems.

**WARNING, use this script AT YOUR OWN RISK.**

## SYNOPSIS

**cfg2html** [**-oshxOcSTflkenaHLvhpPA210**] *option...*

## DESCRIPTION

**cfg2html** collects the system configuration details into a webpage and a text file. **cfg2html** is the âSwiss army knifeâ for sysadmins. It was written to get the necessary information to plan an update, to perform basic troubleshooting or do performance analysis.

This includes the collection of Cron and At jobs, installed Hardware, installed Software, Filesystems, Dump- and Swap-configuration, LVM, Network Settings, Kernel, System enhancements and Applications, and Subsystems.

If you like, you can add a description of the system into **/etc/cfg2html/systeminfo**

example:

Main database server using special software.

Contact: Micky Mouse <admin@foo.bar>

Location: Mars Moon

and so on.

If file **/etc/cfg2html/files** exists, you can add your own list of files to collect - see example included.

## Switches

The following switches are available from the command line:

- o** *dir*  
Override the default Logfile output location for the generated HTML and text output (e.g. **/var/log/cfg2html**) to the specified directory  
Alternatively, you can specify the location of the logfile by setting the environment variable **OUTDIR="/path/to/dir"**  
Specifying the **-o** option supercedes the definition of the OUTDIR environment variable
- 0** append the datestamp to the Logfile (OUTDIR) with format Day-Month-Year-HourMinute, e.g. 28-Oct-2020-2212
- 1** append the datestamp to the Logfile (OUTDIR) with format Day-Month-Year, e.g. 28-Oct-2020
- 2** append the datestamp to the Logfile (OUTDIR) with the date format of your own choosing. e.g specifying **%Y%m%d-%H%M** would append the date with a format of 20201028-2212 to the log file name.
- v** output cfg2html version information, kernel information and exit

**-h** output usage information and exit

By default, information in each of the following categories will be collected. Use the following options to disable the collection of the indicated configuration information:

**-s** disable: System  
**-c** disable: Cron  
**-S** disable: Software  
**-f** disable: Filesystem  
**-l** disable: LVM  
**-k** disable: Kernel  
**-e** disable: Enhancements  
**-n** disable: Network  
**-a** disable: Applications  
**-H** disable: Hardware  
**-x** disable listing the executable files in the PATH  
 This switch has no effect if -s is specified.  
**-O** disable listing open files that have been deleted  
 This switch has no effect if -s is specified.  
**-A** disable collecting the Altiris ADL Agent files, if present

Other options

**-p** enable collecting the system's Log files. If running on an HP PROLIANT SERVER, collect additional relevant logs.  
**-P** enable the processing of the custom plugins that you may have configured  
**-L** disable showing the command that was executed to generate the html output in each section  
**-T** enable showing performance information regarding the execution of cfg2html

## VARIABLES

### OUTDIR

defines the folder where the generated reports are saved. The -o switch overrides the setting of this environment variable.

A useful technique is to specify it on the command line when executed. E.g.

OUTDIR=/path/to/dir ./cfg2html

Note that using the -0 -1 or -2 switches append the indicated date stamp to the log files created in OUTDIR.

### LOCALPATH

If not specified, cfg2html will by default display all of the executable files in the deemed to be optimal PATH as stated in

<http://security.stackexchange.com/questions/117535/ordering-of-the-path-environment-variable>

defined at the beginning of the path, followed by specific additional directories that cfg2html has found to be installed.

**LOCALPATH** provides a means for the Sysadmin to specify the PATH for which they want to see the contents. If defined, the variable should be set to a colon-separated list of directories, which do

not need to exist.  
 e.g. LOCALPATH=/operator/bin:/sbin:/bin:/usr/sbin:/usr/bin  
 The best location for setting this variable is in  
 /etc/cfg2html/local.conf  
 but like **OUTDIR**, can also be specified on the command line, if so desired.

## FILES

After installation, the following files will exist:

### **/etc/cfg2html/systeminfo**

The system description you can update

### **/etc/cfg2html/files**

The contents of the files listed will be collected into the report

### **/etc/cfg2html/local.conf**

Can contain local switch over-rides and local variable definitions

After cfg2html is run, the following files will be generated in OUTDIR:

### **<hostname>.html**

the generated documentation in HTML form

### **<hostname>.txt**

the generated documentation in ASCII form

### **<hostname>.err**

error logfile (if any errors are generated)

### **<hostname>.partitions.save**

Disk Partitions to restore from

### **<hostname>.\_xpio.csv**

(only on systems connected to HP Disk Arrays): the output of the xpinfo command

Other files:

### **README**

Additional documentation and installation instructions.

### **changelog**

Revision history for cfg2html and associated files. Also, see the git log.

## SEE ALSO

**README**, **License.txt** and **changelog** files in the source directory.

**NOTES**

Only **root** can run the **cfg2html** wrapper script.

For security reasons, it is best to store the HTML and ASCII files in a safe place where only the root user has access, or remove the files from your file system.

**CONTACTING**

If you like the script, want to learn more about it, or want to see a feature added to it that isn't already here, then please subscribe to the project mailing list. The project web page is at **<http://www.cfg2html.com>** and the source code is hosted on github at: **<https://github.com/cfg2html/cfg2html>**

**WARNINGS**

use the **cfg2html** script at your own risk! See the file License.txt for details!

**DIAGNOSTICS**

**cfg2html** writes errors to <hostname>.err

**BUGS**

There are probably a lot of bugs. We are currently using this script successfully for our own systems, but there are many features included that we don't have any way to test right now. If you find a bug or have a comment or suggestion about the script, please submit an issue at <https://github.com/cfg2html/cfg2html/issues> or you can also email bugs and issues to the authors. You can find the address in the *AUTHORS* section. We'd love to hear from you. :-)

**AUTHORS**

Original cfg2html HP-UX version written by Ralph Roth <cfg2html@hotmail.com> and ported to Linux by Michael Meifert, maintained by Gratien D'haese and Ralph Roth on github.com.

See the source code and ChangeLog file for the complete history and credits.



**NAME**

checkopt – check used options in shellscripts

**SYNOPSIS**

**edrc/lib/checkopt -h**

**checkopt** "*used\_opts*" "*truthtable*"

```
checkopt "used_opts" "header
row_1
row_2
.....
row_n
"
```

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

check if a list of used options matches against a table of allowed option combinations.

This command is intended to be used to efficiently parse arguments in a script.

**OPTIONS**

"*used\_options*"

continuous or space separated string of options that were used. The characters `._:;=+|` are considered as separators and therefore cannot be part of a list of used options.

The order of the options provided in *used\_options* is not significant.

"*truthtable*"

truth table of the format "*header row\_1 row\_2 row\_3 ... row\_n* " :

*header*

the header line defines the meaning of the fields defined in the row settings. Only single-character options can be checked by the **checkopt** command.

The characters `._:;=+|` can be used to format the *header* content.

*row* in the *row* settings the possible correct option combinations are defined. The following settings are recognized as 'option is used': *I*, *T*, *True*, *Y*, *Yes*, the settings *O*, *F*, *False*, *N*, *No* are recognized as 'option not used'.

The *used\_options* are checked against each *row* in the *truthtable*. If the used options

match against a *row*, the **checkopt** command exits with exit code **0**, if no *row* matches, the exit code is **1**.

The characters `._::;=+|-` can be used to format the *row* content.

## ENVIRONMENT

-

## EXIT STATUS

- 0**        used option combination is correct.
- 1**        used option combination is not correct.
- 4**        usage printed.

## FILES

-

## EXAMPLES

### 1) example usage in a shellscript

Parsing of a command having the usage:

**my\_script -h | -f *file* [ -n | -v ]**

```
#!/bin/sh
:
:
parse_args() { args=$*
 debug=False
 Patchfile=""
 Noexecute=False
 Verbose=False

 getopt xhf:n $* > /dev/null 2>&1
 if [$? -ne 0]; then
 usage
 fi

 opts=""
 while [$# -ge 1]; do
 opts="$opts $1"
 case $1 in
 -x) debug=True ;;
 -f) Patchfile=`rel2abs $2`
 shift ;;
 esac
 done
```

```

 -n) Noexecute=True ;;
 -v) Verbose=True ;;
 -h|*) usage ;;
 esac
 shift
done

checkopt "$opts" "
 fnv
 100
 101
 110
 " || usage
} # parse_args

main() {
 parse_args $*
 :
 :
} # main
main $*

```

## 2) alternate checkopt usage

alternate use of **checkopt**. This usage is equal to example 1).

```

checkopt "$opts" "
 fnv
 YNN
 YNY
 YYN
"

```

## 3) alternate checkopt usage

alternate use of **checkopt**. This usage is equal to example 1).

```

checkopt "$opts" "
 fnv
 TFF
 TFT
 TTF
"

```

## 4) alternate checkopt usage

alternate use of **checkopt**. This usage is equal to example 1).

```

checkopt "$opts" "
 f-----n-----v-----
"

```

```
True--False-False
True--False-True
True--True--False
"
```

**SEE ALSO**

**edrcintro(1)**, **getopt(1)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

checkopt was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

setopt, checkopt, CHECKOPT – check used command line options

**SYNOPSIS**

```
#include "checkopt.h"

char *setopt(char opt);

bool checkopt(char *head, int *table);

bool CHECKOPT(char *head, int_1, int_2, ..., int_n);
```

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

**checkopt.h** provides functionality to efficiently check the correct use of command line switch combinations.

The definition of correct command line option combinations is done by a binary truth table that is more easy to maintain than combinations of logical and/or expressions.

**setopt()**

record all used command line options.

Example:

```
#define USAGE "Usage: mycmd [-l | -m][-z][-n | -f filename]\n"

int opt;
while ((opt = getopt(argc, argv, "f:nlmz")) != -1) {
 switch (opt) {
 case 'f':
 file = true;
 filename = strdup(optarg);
 break;
 case 'n':
 file = false;
 break;
 case 'l':
 low = true;
 break;
 case 'm':
 medium = true;
 break;
 case 'z':
```

```

 zero = true;
 break;
 case '?':
 printf(USAGE);
 return 4;
 break;
 }
 setopt(opt);
 }
 for (int i=optind; i<argc; i++){
 printf(USAGE);
 return 4;
 }

```

**checkopt()**

check all options recorded by **setopt()** against a truth table.

The *header* is a string with all options to be checked in the truth table.

In the truth *table* (=integer array) all allowed (good) option combinations are set by setting the related bit in the integer value list in the same order as the options given in the *header* string.

The last entry in the truth *table* needs to be set to **-1**.

Example:

```

int table[] = {
 0b00000000,
 0b00000100,
 0b00001000,

 0b00000001,
 0b00000101,
 0b00001001,

 0b00000010,
 0b00000110,
 0b00001010,

 -1
};

if (!checkopt("mlnf", table)){
 printf(USAGE);
 return 4;
};

```

**CHECKOPT()**

check all options recorded by **setopt()** against a truth table.

The *header* is a string with all options to be checked in the truth table.

In the truth *table* (=integer array) all allowed (good) option combinations are set by setting the related bit in the integer value list in the same order as the options given in the *header* string.

The **CHECKOPT** macro is suited to more conveniently define the truth *table*.

Example:

```

if (!CHECKOPT (
 "mlnf",
 0b00000000,
 0b00000100,
 0b00001000,

 0b00000001,
 0b00000101,
 0b00001001,

 0b00000010,
 0b00000110,
 0b00001010
)) {
 printf(USAGE);
 return 4;
};

```

## RETURN VALUE

the return value of **checkopt()** and **CHECKOPT()** is **true**, if the option combination is correct (respectively found in the truth table) and **false** if the option combination is not correct (respectively not found in the truth table).

## ENVIRONMENT

-

## FILES

lib/\$OSID/includes/checkopt.h

## EXAMPLES

-

## SEE ALSO

edrcintro(1), **checkopt**(3), <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>, **osid**(3), **program.h**(3), **strings.h**(3), **utility.h**(3), **wa2lc**(3)

**NOTES**

-

**BUGS**

The truth *table* can handle up to 31 options.

**AUTHOR**

checkopt.h was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.



**NAME**

choice – prompt for a key to be pressed

**SYNOPSIS**

**edrc/lib/choice** "*prompt*" *keylist* [ *default\_key* [ *options* ] ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

This command is used in scripts to prompt a user to press a key.

As long as a key is pressed that is not specified as valid in the *keylist* the **choice** prompt is repeated.

It is also possible to specify a default key that takes in effect when <ENTER> or <RETURN> without entering a key is pressed.

All input is logged to a logfile if the environment variable **\$EDRC\_LOGFILE** is set and the **NO\_LOG** option isn't set.

**OPTIONS**

*prompt*      prompt to be displayed.

*keylist*      a string of keys which are accepted by **choice**. If a key is entered the **choice** prompt is re-displayed and the user is requested to enter a valid key. The *keylist* is case-sensitive.

*default\_key*      default key. This key will be returned if the <ENTER> key is pressed.

*options*

**NOT\_NULL**

null input is not accepted.

**NO\_LOG**

do not record the input to the logfile.

**LOG\_STARS**

log a star (\*) to the logfile instead of the real input.

**ENVIRONMENT****\$EDRC\_LOGFILE**

the accepted input read by **choice** is logged to the file specified in this environment variable (if set).

**EXIT STATUS**

- 0** no error.
- 4** usage listed.

**EXAMPLES**

The following examples are script cut-outs of Bourne-, Korn- or Bash shell scripts:

**1) Yes/No user prompt**

```
answer=`choice "remove file? <yn>" yn n`
if [$answer = y]; then
 \rm -f $file
 echo "File removed"
else
 echo "File not removed"
fi
```

**2) A menu**

```
cat <<EOM
p) print user credentials
a) create user account
h) create home directory

q) quit
EOM

ans=`choice "?" pahq q`
case $ans in
p) print_users ;;
a) create_account ;;
h) create_homedir ;;
q) exit ;;
esac
```

**3) A menu including logging all actions**

```
Logfile=/var/log/user_management.log

EDRC_LOGFILE=$Logfile; export EDRC_LOGFILE

{
```

```

cat <<EOM
p) print user credentials
a) create user account
h) create home directory

q) quit
EOM

ans=`choice "?" pahq q`
case $ans in
p) print_users ;;
a) create_account ;;
h) create_homedir ;;
q) exit ;;
esac
} >> $Logfile

```

#### 4) Run a command until 'n' is pressed

```

while [`choice "count? <yn>" yn y` = y]; do
 echo "number of processes: `ps -ef | wc -l`"
done

```

#### SEE ALSO

**edrcintro(1), input(3), sh(1), ksh(1)**

#### NOTES

-

#### BUGS

-

#### AUTHOR

choice was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

#### COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

cltrash – tidy up global trash and \*.TRASHED files

**SYNOPSIS**

**edrc/bin/cltrash** [ **-h** ]

**cltrash -a purge** [ **-d days** ]

**cltrash -a remove** [ **-d days** ] [ **-r rootdir** ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

The **cltrash** command is used to tidy up the global trash and the file systems after the use of the **trash** command.

To tidy up the global trash properly the **cltrash** command supports two actions:

**purge**        purge (remove) files which were trashed a certain number of days ago from the trash.

**remove**      remove the *filename.TRASHED* "information and instruction files" which are generated a certain number of days ago from the system.

Therefore the **cltrash** command has to be started once to purge the global trash with the **purge** action and once on each system participating on the global trash with the **remove** action to remove the related "information and instruction files" from the system. If you don't, the user thinks that a file can still be retrieved from the trash, even it has been purged or a trashed file still uses disk space even if the "information and instruction file" does not exist and the user has no hint any more that the file could still be retrieved.

**OPTIONS**

**-h**            usage message.

**-a purge**     purge (remove) files which were trashed a number of days, optionally to be specified with the **-d** option, ago from the trash.

**-a remove**    remove the *filename.TRASHED* "information and instruction files" which are generated a number of days, optionally to be specified with the **-d** option, ago from the system.

**-d days**      number of days. If not specified this option defaults to 30 days.

**-r rootdir**    start dir to search for *filename.TRASHED* files on the system. If not specified this option defaults to /.

**EXIT STATUS**

- 0** no error.
- 1** LOGDIR or TRASHDIR does not exist. See **trash.cfg**(4) if you get this error.
- 2** operating system is not supported. See **osid**(3) if you get this error.
- 3** rootdir does not exist.
- 4** usage listed.
- 6** configuration file **edrc/etc/trash.cfg** does not exist.

**FILES**

**edrc/etc/trash.cfg**  
configuration file for **trash**, **untrash** and **cltrash**.

**edrc/var/trash/log/**  
default logfile location of **trash** , **untrash** and **cltrash**.

**edrc/var/trash/files/**  
default location for trashed files.

*filename*.**TRASHED**  
information and instruction file to document the file trashing and to give instructions how to recover ( **untrash** ) the trashed file.

**EXAMPLES**

- 1) automated tidy up of a global trash using cron

In this example the **cron** entry of all participating systems is identical and the global trash is physically located in the directory **/data\_adm1/dat/trash**. This is why the **purge** action is only executed if the directory exists on the system (important to check in a clustered environment where the global trash should resist on a cluster package, too).

## 1.1) trash.cfg configuration file

```
#
trash.cfg - trash/untrash/cltrash configuration file
#
[00] 15.02.2005 CWa Initial Version
#

#
Fileformat:
#
OPTION=<VALUE>
#
```

```
Trash basedir. This directory holds all trashed files.
#
TRASHDIR=/dat/trash/files

Logdir.
#
LOGDIR=/dat/trash/logs
```

#### 1.2) needed cron entries to purge the global trash and remove the trash info files

```
[/data_adm1/dat]
[root@adm_db1_prod][ksh]: crontab -l

:
#
purge trash (see also: cltrash -a remove)
#
0 2 * * * [-d /data_adm1/dat/trash] && /opt/edrc/bin/cltrash \
-a purge -d 14 > /dev/null 2>&1
:

:
#
remove trash info files (see also: cltrash -a purge)
#
0 2 * * * [-x /opt/edrc/bin/cltrash] && /opt/edrc/bin/cltrash \
-a remove -r / -d 14 > /dev/null 2>&1
:
```

#### SEE ALSO

**edrcintro(1)**, **trash(1)**, **trash.cfg(4)**, **untrash(1)**

#### NOTES

**cltrash** travels thru all filesystems if the rootdir is specified as / ( **-r** / ), therefore if NFS mounted filesystems are used, those filesystems are searched multiple times.

#### BUGS

-

#### AUTHOR

cltrash was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

cmdlist – list all WA2L/edrc commands

**SYNOPSIS**

**edrc/bin/cmdlist** [ **-h** ]

**cmdlist** [ **-l** ] [ *selection* ]

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

list all commands available in the WA2L/edrc package providing additional credential information.

A user of the WA2L/edrc package is advised to use the **usage**(1) command to get a list of all commands with a short usage.

The **cmdlist** command is intended to be used by a WA2L/edrc developer needing a quick information about implementation details.

Beside **cmdlist**, see also the **binprobe** command to check which binaries (**TYPE**: **BIN**, **JBIN**, **LSCR**, **PSCR**, **YSCR**) are able to run on the operating system where currently logged in.

The output of **cmdlist** consists of a table having the following columns:

**COMMAND**

*command* executable name.

**TYPE** indicator of *command* realization (the same **TYPE** indicator is used in the **compatibility**(1) manual page):

**BIN** machine binary file. A compiled source code.

**JBIN** Java binary files. A compiled \*.**java** source code.

**SCR** Bourne shell script.

**KSCR** Korn (or Bourne-Again) shell script.

**LSCR** Lua script or program.



|             |                                                                                              |
|-------------|----------------------------------------------------------------------------------------------|
| <b>PSCR</b> | Perl script.                                                                                 |
| <b>YSCR</b> | Python script.                                                                               |
| <b>LINK</b> | symbolic link to an other <i>command</i> providing an alternate name for an executable file. |

## DIRECTORY

directory in which the *command* is located (see also: **edrc**(1m)):

### **bin** (=edrc/bin/)

executables planned to be used mostly from the command line.

This directory is in **\$PATH** when using **edrc**, **shell** or **eshell**.

### **lib** (=edrc/lib/)

executables planned to be used mostly from within scripts because this *commands* often have the character of supporting functions and need often more initialization.

This directory is in **\$PATH** when using **edrc**, **shell** or **eshell**.

### **sbin** (=edrc/sbin/)

executables planned to be used from the command line.

This directory is not in **\$PATH** when using **edrc**, **shell** or **eshell**.

### **pbin** (=edrc/pbin/)

executables to be used mostly from the command line as normal (unprivileged) user but with elevated permissions. Depending on the system, *commands* in this directory use automatically **sudo** or **pfexec** to elevate the permissions without the need to manually call **sudo command** or **pfexec command**.

This directory is not in **\$PATH** when using **edrc**, **shell** or **eshell**, but should be added to the **\$PATH** of the normal (unprivileged) user.

## STARTMODE

mode how the *command* is started. In general it can be distinguished between tree categories. However the user does not note a difference (and does not need to know the **STARTMODE**) when using the *commands* of any of the categories (sometimes such a behaviour is called "transparent"):

**direct** direct start of a command where **TYPE** is **SCR**.

*command* a command is started thru an alternate name (**TYPE** is **LINK**) provoking a different behaviour.

This method of calling the same command thru different names is a common concept on UNIX like systems to save implementation effort.

**.name\_wrapper**

start thru a wrapper (see related *name\_wrapper*(1) manual page), where before the real start additional checks are executed and specific settings are automatically applied:

**.os\_wrapper**

start of operating system dependent binaries.

**.java\_wrapper**

start Java binary code using a system-wide installed Java RE (JRE, Java Runtime Environment).

Note: There is no JRE distributed/bundled with WA2L/edrc and needs to be installed somewhere on the system.

**.perl\_wrapper**

start Perl script using a system-wide installed Perl interpreter.

**.python\_wrapper**

start Python script using a system-wide installed Python interpreter.

**.ksh\_wrapper**

start Korn- or Bourne-Again (**bash**) script using **ksh** or **bash** depending on availability on the system.

**.pf\_wrapper**

start executable using **sudo** or **pfexec** depending on availability/configuration on the system.

**.daemon\_wrapper**

start a daemon-like executable.

**daemon** start a daemon-like executable exceptionally not using the **.daemon\_wrapper**.

## OPERATINGSYSTEM

a list of operating systems (**osid**) for which distinct binaries are provided/bundled in/with WA2L/edrc.

This does not necessarily mean, that on operating systems for which the WA2L/edrc does not provide a binary file, the related command cannot be started from within WA2L/edrc.

For a description of this concept in WA2L/edrc, see **os\_wrapper**(1) manual page.

OPTIONS

- h usage message.
- l provide list as CSV file format.

ENVIRONMENT

-

EXIT STATUS

- 0 no error.
- 4 usage displayed.

FILES

-

EXAMPLES

1) example cmdlist output:

| COMMAND            | TYPE | DIR  | STARTMODE     | OPERATINGSYSTEM                           |
|--------------------|------|------|---------------|-------------------------------------------|
| -----              | ---- | ---- | -----         | -----                                     |
| ansi2txt           | BIN  | lib  | .os_wrapper   | Linux<br>Linux-32<br>Linux-64             |
| appendpdf          | PSCR | lib  | .perl_wrapper | *                                         |
| apply2file         | SCR  | bin  | direct        | *                                         |
| apply2sw_inventory | SCR  | bin  | direct        | *                                         |
| apprevision        | SCR  | lib  | direct        | *                                         |
| approot            | SCR  | lib  | direct        | *                                         |
| apropos            | SCR  | bin  | direct        | *                                         |
| asup               | SCR  | pbin | .pf_wrapper   | *                                         |
| asup               | LINK | bin  | sat           | *                                         |
| banner             | BIN  | lib  | .os_wrapper   | Linux<br>Linux-32<br>Linux-64<br>Linux-ia |
| bget               | PSCR | lib  | .perl_wrapper | *                                         |
| :                  |      |      |               |                                           |
| :                  |      |      |               |                                           |
| expect             | BIN  | lib  | .os_wrapper   | HP-11                                     |

| cmdlist(1m) | Maintenance Commands |     |     |             | cmdlist(1m)                                                                                       |
|-------------|----------------------|-----|-----|-------------|---------------------------------------------------------------------------------------------------|
|             |                      |     |     |             | HP-11i<br>HP-11ia<br>Linux<br>Linux-32<br>Linux-64<br>Linux-ia<br>Solaris<br>Solaris-x86          |
|             | fcreate              | BIN | lib | .os_wrapper |                                                                                                   |
|             | :                    |     |     |             |                                                                                                   |
|             | :                    |     |     |             |                                                                                                   |
|             | zmore                | BIN | bin | .os_wrapper |                                                                                                   |
|             |                      |     |     |             | HP-11<br>HP-11i<br>HP-11ia<br>Linux<br>Linux-32<br>Linux-64<br>Linux-ia<br>Solaris<br>Solaris-x86 |
|             | (804)                |     |     |             |                                                                                                   |

**SEE ALSO**  
 edrcintro(1), binprobe(1m), cmdlist(1m), compatibility(1), daemon\_wrapper(1), java\_wrapper(1), ksh\_wrapper(1), lua\_wrapper(1), osid(3), os\_wrapper(1), perl\_wrapper(1), pf\_wrapper(1), python\_wrapper(1), usage(1)

**NOTES**  
 -

**BUGS**  
 -

**AUTHOR**  
 cmdlist was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

**COPYRIGHT**  
 Copyright © 2021 Christian Walther  
 This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

cmmon – monitor all MC/ServiceGuard cluster packages in a cluster

**SYNOPSIS**

edrc/bin/cmmon

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

monitor of all MC/SG packages on HP-UX systems having MC/ServiceGuard installed. If an interval of 0 is specified a one time output is provided, else the output is continuously.

**OPTIONS**

-

**ENVIRONMENT**

-

**EXIT STATUS**

- |           |                                                                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>0</b>  | no error.                                                                                                                                                                            |
| <b>2</b>  | operating system not supported.                                                                                                                                                      |
| <b>11</b> | temporary directory could not be claimed or created in <b>/tmp</b> . Check the system temporary directory <b>/tmp</b> if you get this error, it is an indicator of system intrusion. |

**FILES**

-

**EXAMPLES**

-

**SEE ALSO**

**edrcintro(1)**, **cmswitch(1m)**, **cmviewcl(1m)**

**NOTES**

This command only runs on HP-UX systems having MC/ServiceGuard installed.

**BUGS**

-

**AUTHOR**

cmmon was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

cmswitch – switch a MC/ServiceGuard cluster package to a node

**SYNOPSIS**

**edrc/bin/cmswitch**

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

switch a MC/SG package to a node. Using this interactive command is much faster then entering all commands needed for a cluster package switch from one to an other node. You can restart a package if you specify the host where the package is currently running as the target node.

Be aware that the **cmswitch** command does not check if it is possible to switch the package to the specified target node, therefore you should check the possible package switch positions first using the **cmmon** command.

This command is ideal to be used while efficiently testing the correct functionality of the configuration of the cluster package configuration because also timing information is printed. Therefore the command output is well suited for test protocol usage.

**OPTIONS**

-

**ENVIRONMENT**

-

**EXIT STATUS**

**0** no error.

**4** operating system not supported.

**FILES**

**edrc/var/log/cmswitch.log**

log file of the **cmswitch** command.



**EXAMPLES**

-

**SEE ALSO**

**edrcintro(1)**, **cmhaltpkg(1m)**, **cmmodpkg(1m)**, **cmmon(1m)**, **cmrunpkg(1m)**

**NOTES**

This command only runs on HP-UX systems having MC/ServiceGuard installed.

**BUGS**

-

**AUTHOR**

cmswitch was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

comm – compare two sorted files line by line

**SYNOPSIS**

**comm** [*OPTION*]... *FILE1 FILE2*

**DESCRIPTION**

Compare sorted files *FILE1* and *FILE2* line by line.

When *FILE1* or *FILE2* (not both) is –, read standard input.

With no options, produce three-column output. Column one contains lines unique to *FILE1*, column two contains lines unique to *FILE2*, and column three contains lines common to both files.

**-1** suppress column 1 (lines unique to *FILE1*)

**-2** suppress column 2 (lines unique to *FILE2*)

**-3** suppress column 3 (lines that appear in both files)

**--check-order**

check that the input is correctly sorted, even if all input lines are pairable

**--nocheck-order**

do not check that the input is correctly sorted

**--output-delimiter=STR**

separate columns with *STR*

**--total** output a summary

**-z, --zero-terminated**

line delimiter is NUL, not newline

**--help** display this help and exit

**--version**

output version information and exit

Note, comparisons honor the rules specified by 'LC\_COLLATE'.

**EXAMPLES**

comm -12 file1 file2

Print only lines present in both file1 and file2.

comm -3 file1 file2

Print lines in file1 not in file2, and vice versa.

**AUTHOR**

Written by Richard M. Stallman and David MacKenzie.

**REPORTING BUGS**

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>

Report comm translation bugs to <<http://translationproject.org/team/>>

**COPYRIGHT**

Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<http://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

**SEE ALSO**

join(1), uniq(1)

Full documentation at: <<http://www.gnu.org/software/coreutils/comm>>  
or available locally via: info '(coreutils) comm invocation'

NAME

compatibility – WA2L/edrc command compatibility overview (compatibility matrix)

SYNOPSIS

–

AVAILABILITY

WA2L/edrc

DESCRIPTION

This matrix shows for which OSID or group of OSIDs a command has been developed or is available.

COMPATIBILITY MATRIX

All commands of type **BIN** are started thru the **os\_wrapper(1)**. If such a command is not distributed with the WA2L/edrc package for a certain OSID, but it is available on the operating system, it will be started, and is therefore also available for an OSID not listed here. This is often the case for commands that are installed by standard on certain operating systems only.

The *X* entry for a command indicates for which OSID it is able to run, respectively on which operating system ID the command has been tested.

The *e* entry for a command indicates for which OSID it is expected to run, but has not been tested.

Always use the **binprobe(1m)** command to test binaries and Perl programs.

., A

| COMMAND            | TYPE | * | HP.* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|--------------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| .daemon_wrapper    | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .java_wrapper      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .ksh_wrapper       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .lua_wrapper       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .os_wrapper        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .perl_wrapper      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .pf_wrapper        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| .python_wrapper    | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ansi2txt           | BIN  | – | –    | –     | –     | –      | –       | X       | X     | –        | X        | –        | –       | –           | –      |
| appendpdf          | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | –      |
| apply2file         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| apply2sw_inventory | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| apprevision        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| aproot             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| apropos            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| asup               | SCR  | – | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | –      |

**B, C, D, E**

| COMMAND           | TYPE | * | HP-* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|-------------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| banner            | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| batteryalert      | SCR  | - | -    | -     | -     | -      | -       | X       | X     | X        | X        | -        | -       | -           | -      |
| binprobe          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| bget              | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| bunzip2           | SCR  | - | -    | -     | X     | X      | X       | -       | X     | -        | X        | -        | X       | X           | -      |
| busy              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| bwcreate          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| bzcat             | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| bzdifff           | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| bsgrep            | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| bzip2             | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| bzip2recover      | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| bzmore            | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| catcomp           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ccrypt            | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| cfg2html          | SCR  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| CGI               | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| changepwd_trusted | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | -       | -           | -      |
| checkopt          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| choice            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| cltrash           | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| cmdlist           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| cmmon             | SCR  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | -       | -           | -      |
| CMS::MediaWiki    | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| cmswitch          | SCR  | - | X    | X     | X     | X      | X       | -       | -     | -        | -        | -        | -       | -           | -      |
| connect           | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | -       | -           | -      |
| consolidate       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| cpanm             | LSCR | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| cpio              | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | -       | -           | -      |
| crfile            | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| crond             | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| cronhandler       | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| crypt             | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| csv               | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| csvcat            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| csv2worksheet     | JBIN | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| csvq              | BIN  | - | -    | -     | -     | -      | -       | -       | X     | -        | X        | -        | -       | -           | -      |
| days              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| DBD::SQLite       | PM   | - | -    | -     | -     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| DBI               | PM   | - | -    | -     | -     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| dbrep             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| dig               | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | -       | -           | -      |
| directories       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| dmidecode         | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| du_report         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| dos2ux            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| duvi              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ecrontab          | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| ecronnext         | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| edrc              | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| edrcenv           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| edrcinit          | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| edrcman           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| edrcpack          | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| edrcroot          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| envpasswdstrip    | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| epub2pdf          | JBIN | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| eshell            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| eterm             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| exiftool          | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| expect            | BIN  | - | -    | -     | X     | X      | X       | -       | X     | -        | X        | -        | X       | X           | -      |

**F, G, H, I, J**

| COMMAND          | TYPE | * | HP-* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|------------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| fcreate          | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | -           | -      |
| fields2swvi      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| File::Tail       | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| filedist         | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| filegrep         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| filelink         | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| filesize         | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| filewatch        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| fit2with         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| fnmatch          | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| fnmatch          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| freespace        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| fssum            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ftps             | JBIN | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| gecos            | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| getfilesystem    | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| gethostbyname    | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| getmountpoint    | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| gid              | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| glibc.version    | BIN  | - | -    | -     | -     | -      | -       | X       | X     | -        | X        | -        | -       | -           | -      |
| gnutar           | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| group            | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| gs               | BIN  | - | -    | -     | -     | -      | -       | -       | X     | -        | X        | -        | -       | -           | -      |
| gunzip           | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| gzip             | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| h2               | JBIN | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| histlist         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| homedir          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| hostaliases      | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| hostlist         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| hostlistdat2cfg  | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| HTML::Parser     | PM   | - | -    | -     | -     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| html2mht         | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| hwinventory      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ident            | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| indent           | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| info             | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | -       | X           | -      |
| ini.bash         | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| ini.csh          | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| ini.ksh          | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| ini.sh           | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| input            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| input_targets    | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ipcalc           | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| ipsort           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_config_byhand | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| is_existing      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_osid          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_permitted     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_running       | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_up            | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_user          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_weekend       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| is_writeable     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| java             | BIN  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| job              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| joblog           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| jobstart         | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| jobwatch         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| jq               | BIN  | - | -    | -     | -     | -      | -       | X       | X     | -        | X        | -        | X       | -           | -      |

**K, L, M, N**

| COMMAND        | TYPE | * | HP-* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|----------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| kalc           | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| kshell         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lbanner        | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| lcat           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| leo            | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| lgrep          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lgcheckd       | PSCR | X | e    | e     | e     | e      | e       | X       | X     | X        | X        | e        | e       | e           | e      |
| lgcpattern     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| listtemp       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ll             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| llcomp         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| locate         | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| locations      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| log            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| logcat         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| logcheckd      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| logcut         | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| logrotate      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| loggrep        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| logtail        | KSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| logview        | KSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lots           | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| lotsctl        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| lscmp          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lscp           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lsmv           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lsof           | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| lspm           | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| lua            | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| luac           | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| luarocks       | LSCR | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| luaenv         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| luaversion     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| lynx           | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| mail_file      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| maketemp       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| makeuser       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| man            | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| man2html       | BIN  | - | -    | -     | X     | X      | -       | -       | X     | X        | X        | -        | X       | X           | -      |
| mc             | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| mcedit         | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| mcview         | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| md2html        | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| md5sum         | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| md5string      | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | -        | X        | X       | X           | -      |
| mediawiki      | PSCR | - | -    | -     | -     | -      | X       | -       | X     | -        | X        | -        | X       | -           | -      |
| MediaWiki::API | PM   | - | -    | -     | -     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| MIME::Base64   | PM   | - | -    | -     | -     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| MIME::Lite     | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| mkuser         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| msg            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| name           | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | -        | X       | -           | -      |
| nano           | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| nginx          | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| nmap           | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| nologin        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| nping          | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |

**O, P, Q, R**

| COMMAND            | TYPE | * | HP-* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|--------------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| omniutil           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| osid               | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| osid.probe         | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| osup               | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| outex              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pack               | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| passwdcombine      | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| passwdsort         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| passwdsyncd        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| passwdsyncd_apply  | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| patchinstall       | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| pdfmetaedit        | JBIN | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pdfscissors        | JBIN | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| perlenv            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| perlversion        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pid                | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pip                | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pkg_hostname       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pkzip              | BIN  | - | -    | -     | X     | X      | X       | -       | X     | -        | -        | -        | X       | -           | -      |
| pl                 | BIN  | - | -    | -     | -     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| pmdesc             | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| pod2html           | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| portscan           | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| print_header       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| print_index        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| print_list         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pscount            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| psjoin             | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| pslist             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ps2pdf             | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| pstree             | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| psup               | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| purgetemp          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pwcrypt            | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| pwsafe             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pythonenv          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| pythonversion      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| random             | KSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| rcat               | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rcomm              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rcmd               | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rdiff              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| readline           | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | X       | X           | -      |
| rel2abs            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| remote_copy        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| remote_shell       | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| removetemp         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| repeat             | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| resolve_targetlist | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| revision           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rl                 | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rnano              | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| role_option        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| rosid              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rsat               | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| rsash              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| rsync              | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | X        | -        | X       | X           | -      |

## S, T

| COMMAND            | TYPE | * | HP-* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|--------------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| sat                | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sav                | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| Schedule::Cron     | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| screen             | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| scriptextract      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| scriptgrep         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| scriptheadersync   | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| scriptmenupath     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| scriptrevision     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| scriptsequence     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| scripttitle        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| seconds            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| sectioncat         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| select_columns     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| server_environment | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| shell              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| shellinaboxd       | BIN  | - | -    | -     | -     | -      | -       | -       | X     | -        | X        | -        | -       | -           | -      |
| shlib              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| sortc              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| sparse             | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| splitvt            | BIN  | - | -    | -     | X     | X      | X       | -       | X     | -        | X        | -        | -       | -           | -      |
| sqlite             | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| ssh-keyadd         | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ssh-exec           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| stat               | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| strace             | BIN  | - | -    | -     | -     | -      | -       | -       | X     | X        | X        | -        | -       | -           | -      |
| streamcat          | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| svi                | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sw_report          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| swvi               | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| symlink            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| sys                | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sysconfig          | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| syscp              | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sysdiff            | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sysinfo            | BIN  | - | -    | -     | X     | X      | X       | -       | -     | X        | X        | -        | X       | -           | -      |
| syspoll            | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sysvi              | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| sys2html           | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| tcpdump            | BIN  | - | -    | -     | X     | X      | X       | -       | -     | -        | -        | -        | X       | X           | -      |
| Text::vCard        | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| textblock          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| textcolor          | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| thttpd             | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| timer              | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| timezone           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| title              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| Tk                 | PM   | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| today              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| tolower            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| tomorrow           | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| top                | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | -        | -        | X       | -           | -      |
| toupper            | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| tpl                | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| traceroute         | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | -        | -        | X       | -           | -      |
| trash              | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| truss              | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | -           | -      |
| tscat              | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| tscat              | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | -        | X        | X       | X           | -      |
| tsize              | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| tty_columns        | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| tty_variable       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ttyplay            | BIN  | - | -    | -     | X     | X      | X       | -       | X     | -        | X        | -        | -       | -           | -      |
| ttyrec             | BIN  | - | -    | -     | X     | X      | X       | -       | X     | -        | X        | -        | -       | -           | -      |
| ttytime            | BIN  | - | -    | -     | X     | X      | X       | -       | X     | -        | X        | -        | -       | -           | -      |
| tzdump             | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |



## U, V, W, Y, X, Z

| COMMAND    | TYPE | * | HP-* | HP-10 | HP-11 | HP-11i | HP-11ia | *Linux* | Linux | Linux-ia | Linux-64 | Solaris* | Solaris | Solaris-x86 | Sun-OS |
|------------|------|---|------|-------|-------|--------|---------|---------|-------|----------|----------|----------|---------|-------------|--------|
| uid        | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| uncgz      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| uncczip    | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| unczst     | SCR  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| unczstd    | SCR  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| undeb      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| unigpath   | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| updatedb   | SCR  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| unrpm      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| untgz      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| untrash    | KSCR | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| untzst     | SCR  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| untzstd    | SCR  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| unzip      | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| usage      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| user       | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| user_info  | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| uencode    | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| uudecode   | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| ux2dos     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| vadsp      | SCR  | - | X    | X     | X     | X      | X       | -       | -     | -        | -        | -        | -       | -           | -      |
| vcats      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vdifff     | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vgrep      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vlist      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vls        | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vmore      | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vpurge     | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vrestore   | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vsav       | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| vsdfml     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| vvvi       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| WA2l::Util | PM   | X | X    | X     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| watchdog   | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| wget       | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| whatls     | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| whereami   | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| wholsin    | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| winexe     | BIN  | - | -    | -     | -     | -      | -       | -       | X     | -        | X        | -        | -       | -           | -      |
| wmic       | BIN  | - | -    | -     | -     | -      | -       | -       | X     | -        | X        | -        | -       | -           | -      |
| woist      | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| xbdf       | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| xlog       | SCR  | - | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| xml2csv    | JBIN | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |
| xml2json   | BIN  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| xmore      | PSCR | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | -      |
| xpid       | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| xtee       | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | -           | -      |
| yesterday  | SCR  | X | X    | X     | X     | X      | X       | X       | X     | X        | X        | X        | X       | X           | X      |
| ypxfr_all  | SCR  | - | X    | X     | X     | X      | X       | -       | -     | -        | -        | X        | X       | X           | -      |
| zcat       | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| zdiff      | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| zgrep      | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| zip        | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| zless      | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| zmore      | BIN  | - | -    | -     | X     | X      | X       | -       | X     | X        | X        | -        | X       | X           | -      |
| zstd       | BIN  | - | -    | -     | -     | -      | -       | -       | -     | -        | X        | -        | -       | -           | -      |

## SEE ALSO

**EDRC**(1), **edrcintro**(1), **binprobe**(1m), **cmdlist**(1m), **daemon\_wrapper**(1), **ksh\_wrapper**(1), **mandir**(4), **osid**(3), **os\_wrapper**(1), **perl\_modules**(1), **perl\_wrapper**(1), **python\_wrapper**(1)

## NOTES

Description of startup of the command **TYPES**:

**BIN** started thru the **.os\_wrapper**. See **os\_wrapper**(1) for more information.

**JBIN** started thru the **.java\_wrapper**. See **java\_wrapper**(1) for more information.

**SCR** direct startup or start thru the **.daemon\_wrapper**, depending on the functionality. See **daemon\_wrapper**(1) for more information.

- PSCR** started thru the **.perl\_wrapper**. See **perl\_wrapper(1)** for more information.
- PM** Perl modules are used using the **use <Module>;** statement in a Perl script. The modules bundled with WA2L/edrc are available when the Perl script is started thru the **.perl\_wrapper**. If a certain bundled Perl module is not available for a certain OS-Id, but it is installed on the system, the module is available for the calling Perl script. See **perl\_wrapper(1)** and **perlenv(3)** for more information.
- KSCR** started thru the **.ksh\_wrapper**. See **ksh\_wrapper(1)** for more information.
- LSCR** started thru the **.lua\_wrapper**. See **lua\_wrapper(1)** for more information.
- YSCR** started thru the **.python\_wrapper**. See **python\_wrapper(1)** for more information.

## BUGS

-

## AUTHOR

compatibility was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

## COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

**connect** – Make socket connection using SOCKS4/5 and HTTP tunnel

**SYNOPSIS**

```
connect [-dnhst45] [-R resolve] [-p local-port] [-w seconds]
 [-H [user@] proxy-server [:port]]
 [-S [user@] socks-server [:port]]
 [-T proxy-server [:port]]
 [-c telnet proxy command]
 host port
```

**DESCRIPTION**

*Connect* is a proxy tool to enable *OpenSSH* and other TCP/IP utilities to run through SOCKS4/5 and HTTP proxy gateways. The arguments *host* and *port* are for the target hostname and port number to connect to.

The **-H** option specifies the hostname and port number of the HTTP proxy server to be used as the relay. If *port* is omitted, the default value of 80 will be used. This value can also be specified using the environment variable HTTP\_PROXY and pass the **-h** option to use it.

The **-S** option specifies the hostname and port number of the SOCKS server to be used as the relay. As with **-H**, the port number can be omitted, in which case the default value is 1080. This value pair can also be specified in the environment variable SOCKS5\_SERVER and use it with the **-s** option.

The **-4** and **-5** options are for specifying the SOCKS relaying and indicates which protocol version to use. The options are valid only when used with **-s** or **-S**. The default is **-5** (protocol version 5).

The **-R** option is for specifying method to resolve the hostname. Three keywords (*local*, *remote* and *both*) or an IP address in dot-notation are acceptable. The keyword *both* means, “Try local first, then remote”. If a dot-notation IP address is specified, use *IP address* as the name server. The default is *remote* for SOCKS5 or *local* for others. On SOCKS4 protocol, remote resolving method (*remote* and *both*) requires a protocol 4a supported server.

The **-p** option will forward a local TCP port instead of using standard input and output.

The **-P** option is the same as **-p** except that it will keep the remote session open. The program will wait the port and hold the remote session without disconnecting. To disconnect the remote session, send EOF to stdin or kill the program.

The **-w** option specifies timeout *seconds* for opening a connection with the target *host*.

The **-d** option is used for debugging connections. If you fail to connect to a host, use this option to check requests to and responses from server.

**EXAMPLES**

You can omit the *port* argument when program name is special format containing port number itself. For example,

```
$ ln -s connect connect-25
```

means this connect-25 command is specifying port number 25 already, thus the second argument is not needed and it will be ignored if specified.

To use proxy, this example is for SOCKS5 connection to connect to *host* at port 25 via SOCKS5 server on *firewall* host.

```
$ connect -S firewall host 25
```

or

```
$ SOCKS5_SERVER=firewall; export SOCKS5_SERVER
$ connect -s host 25
```

For a HTTP-PROXY connection:

```
$ connect -H proxy-server:8080 host 25
```

or

```
$ HTTP_PROXY=proxy-server:8080; export HTTP_PROXY
$ connect -h host 25
```

To forward a local port, for example to use ssh:

```
$ connect -p 5550 -H proxy-server:8080 host 22
($ ssh -l user -p 5550 localhost)
```

When used with OpenSSH, e.g. *ssh*(1), connect can be used with *ProxyCommand* in *ssh\_config*(5) to inter-operate with SOCKS gateways.

Host destinationhost

```
ProxyCommand connect -R local -S proxygw destinationhost %p
```

## ENVIRONMENT

User name for authentication is passed to *connect* in an environment variable and if none defined, the system login name will be used. Similarly, the password can be specified by using an environment variable, external program, specified in **\$SSH\_ASKPASS**, or from tty.

Following environment variables are used for specifying user name. SOCKS: **\$SOCKS5\_USER**, **\$LOGNAME**, **\$USER**

HTTP Proxy: **\$HTTP\_PROXY\_USER**, **\$LOGNAME**, **\$USER**

## LIMITATIONS

Only USER/PASS authentication is supported in SOCKS5 authentication.

Only HTTP Basic Authentication is supported in HTTP proxy authentication.

## SEE ALSO

*ssh*(1), *ssh\_config*(5)

## REFERENCES

SOCKS5 – RFC 1928, RFC 1929, RFC 1961

NEC SOCKS Reference Implementation is available from <http://www.socks.nec.com>.

DeleGate version 5 or earlier can be SOCKS4 server, version 6 can be SOCKS5 and/or SOCKS4 server, and the version 7.7.0 or later can be SOCKS5 and SOCKS4a server. See <http://www.delegate.org/delegate/>.

Hypertext Transfer Protocol HTTP/1.1 – RFC 2616.

HTTP Basic and Digest Authentication – RFC 2617.

**NAME**

consolidate – filter to consolidate unix file content

**SYNOPSIS**

**edrc/bin/consolidate** [ **-h** ]

**consolidate** *structure*

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

filter to consolidate structured unix file content from multiple input.

After using the **consolidate** command, the output should be verified, to ensure it matches to the conventions in your environment.

**OPTIONS**

**-h**           usage message.

*structure*   data structure to consolidate:

**services**

internet network services list as known from the **/etc/services** file.

The consolidated output is sorted by the service number field and not by the service name.

See **services(5)** for more information.

**ENVIRONMENT**

-

**EXIT STATUS**

**0**           no error.

**4**           usage printed.

**FILES**

-

**EXAMPLES****1) consolidate multiple services files**

```
cat services.host1 services.host2 |\
consolidate services > services.out
```

**2) consolidate multiple services files using rcmd**

```
rcmd -n -t host1,host2 -c "cat /etc/services" |\
consolidate services > services.out
```

**SEE ALSO****edrcintro(1), services(4), rcmd(1)****NOTES**

-

**BUGS**

-

**AUTHOR**

consolidate was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

contrib – writing contributed commands for edrc

**SYNOPSIS**

**edrc/contrib/edrc**

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

If you miss a general command in **edrc** that is available in all menus, do not change the **edrc** program.

It is possible to write this commands as a separate program (a contributed command) and place it into the **edrc/contrib/edrc** directory. From then on the new contributed command can be called as if it is an **edrc** internal command. It is then listed in the section **CONTRIBUTED COMMANDS** when the **help** command in **edrc** is invoked.

This will protect you from loosing your work after an upgrade of **edrc**, simplifies the development of additional functionality for **edrc** and will keep the **edrc** command itself stable.

**CONVENTIONS**

A contributed command has to comply with some conventions:

- If the command is limited to be started on certain operating systems only, use **osid(3)** or **is\_osid(3)** to ensure the startup on the designated operating systems only.
- Don't use absolute pathnames. If you have to resolve something within WA2L/edrc, use **approot(3)** and the environment variables provided. See also section **ENVIRONMENT**.
- To save persistent information, use the **\$EDRC\_CONTRIB\_VARDIR** environment variable to do so. You should add the name of the contributed command as prefix to ensure a unique filename.

Example:

```
date > $EDRC_CONTRIB_VARDIR/my_command.begin
```

- If a man page is written for a contributed command, the manpage name convention is **contrib.my\_command.1m** and it has to be placed into the man page directory **man/all/man1m**.
- If a contributed command provides functionalities that can be denied from execution, the **@PROVIDES@** tag has to be used in the file header and the functionality has to be listed.

Example:

```
@PROVIDES@ view display more grep tail head
```

This results in the contributed functionality being listed in the output of the **edrcperm** command within **edrc** in the form **contrib.my\_command.functionality**.

Example:

```
EDRC PERMISSIONS / FUNCTIONALITIES
```

```
All:
 :
 contrib.logs.display contrib.logs.grep contrib.logs.head
 contrib.logs.more contrib.logs.tail contrib.logs.view
 :
```

To handle the denial of a functionality startup in a contributed command, the **is\_permitted(3)** command has to be used.

Example:

```
if [`is_permitted view` = True]; then
 view $log >`tty`
fi
```

## FILE STRUCTURE

A contributed command consists of the following file structure:

*my\_command*

The new contributed command. This file has to have appropriate execute permissions.

*my\_command.nolog*

This is a flagfile. If this file exists the output sent to stdout is not saved to a designated logfile in **var/log**. This file should only be created if the contributed command creates a continuous output that ends only by user intervention. This is to avoid file system fill ups.

## OPTIONS

As all **edrc** commands, the contributed commands do not receive command line options. All options have to be queried by the contributed command using for example the **input(3)** or **choice(3)** commands. See example 2) in the **EXAMPLES** section to see how to query user input.

## ENVIRONMENT

All environment variables exported to a recovery script are exported to a contributed command, too. In addition the **\$EDRC\_CONTRIB\_VARDIR** is exported.

See section **ENVIRONMENT** in the **edrc(1m)** manpage for a complete explanation of all exported environment variables.



## EXIT STATUS

-

## FILES

### **contrib/edrc/**

location of all contributed commands to **edrc**.

### **contrib/edrc/my\_command**

the executable of a contributed command. This file has to have appropriate execute permissions.

### **contrib/edrc/my\_command.nolog**

if this file exists, the output sent to stdout is not logged in a logfile in **var/log**. This file should only be present for contributed commands creating a permanent output to avoid filesystem fill up caused by endless growing logfiles. This is the case if the contributed command monitors a certain system state and only stops monitoring after a user intervention. If the contributed command creates no such continuous output the **my\_command.nolog** should not be created.

### **man/all/man1m/contrib.my\_command.1m**

man page of a contributed command.

## EXAMPLES

See also existing contributed commands in **contrib/edrc**.

### 1) the logtail contributed command

This contributed command provides the **logtail(1)** functionality within **edrc**.

The contributed command **logtail** :

```
#!/bin/sh
#
logtail - start logtail
#
[00] 08.08.2004 CWa Initial Version
[01] 25.01.2006 CWa chg: command name ltail -> logtail
#

test "$DEBUG" = True && set -x

logtail
```

The nolog file **logtail.nolog** (the content of the **logtail.nolog** is not important. **edrc** only checks if the **logtail.nolog** file exists or not):

```
#
logtail.nolog - flag file to prevent output logging
#
```

```
[00] 26.01.2006 CWa Initial Version
#
If this file exists, the output of the relating
contributed command is not logged to a logfile.
#
This make sense if the contributed command
generates continuous output which would generate
very large logfiles.
#
```

## 2) the cmviewcl contributed command

This contributed command provides the **cmviewcl(1)** functionality to **edrc**. **cmviewcl** is a command to display the state of a **MC/ServiceGuard** cluster as available on HP-UX. Therefore it only makes sense to allow the startup of this command on HP-UX.

The contributed command **cmviewcl** :

```
#!/bin/sh
#
cmviewcl - MC/ServiceGuard state
#
[00] 08.02.2003 CWa Initial Version
[01] 18.03.2003 CWa +opt
[02] 10.05.2003 CWa fix: DEBUG
[03] 24.07.2003 CWa +is_osid
#

test "$DEBUG" = True && set -x

is_osid -s $0 -o HP-11,HP-11i || exit $?

opt=`input "options" "" `

msg INFO "MC/ServiceGuard state:"

cmviewcl $opt | more
```

## SEE ALSO

**contrib.edrc(1m)**, **contrib.<contributed\_command>(1m)**, **edrcintro(1)**, **edrc(1m)**, **edrc.cfg(4)**, **edrcenv(1)**, **choice(1m)**, **input(1m)**, **is\_permitted(3)**, **log(3)**

## NOTES

As you might notice already a number of contributed commands are distributed with the WA2L/edrc package. The reason for this is that all operating system dependent commands were moved in an early version of **edrc** to contributed commands and only the core non operating system dependent functionality remained in **edrc**. This should support the concept of **edrc** to be compatible to a wide range of operating systems without modifications.

In addition some additional functionality that does not support the basic **edrc** functionality is first implemented as a contributed command and probably later integrated if all bugs are fixed and the decision is made to include it. In general there is no disadvantage to have some functionality available as contributed commands and not as a core **edrc** command.

The complete list of all contributed commands distributed with WA2L/edrc is given in the man page **contrib.edrc(1m)**.

## BUGS

-

## AUTHOR

contrib was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

## COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

contrib.doc – display recovery script and menu documentation

**SYNOPSIS**

**doc** (contributed WA2L/edrc command)

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

All menu points and menus can be documented. The documentation is displayed using the **doc** contributed **edrc** command.

The documentation is compiled out of commented sections of a recovery script or menu title.

A documentation is structured into several sections similar to Unix man pages. To place text into a documentation section, each line has to be preceded by a section tag consisting of a capital letter and a colon.

To write text for the **DESCRIPTION** section, the recovery script looks like this:

```
:
#
D: This is an example that is displayed as
D: justified text in the DESCRIPTION section
D: of the documentation when displayed
D: with the 'doc' contributed command.
#
:
```

It is not necessary that all your documentation is done at the beginning of a recovery script. Suppose your script conducts several steps; it might make sense that you enter the documentation near the single steps to ensure an update if the code changes in the future.

**PREDEFINED DOCUMENTATION SECTIONS**

The following documentation sections with the given documentation tags are predefined. The tag sequence in the recovery script is not important, the output will be in the sequence as the sections are explained below. Note, that only sections that contain information will be displayed, therefore if you do not enter text for a certain section, it won't be printed.

**NAME**     The name and short description of a menu point as shown when the menu is displayed using the **menu** command in **edrc**. This section is dynamically generated.

**AVAILABILITY**

This section contains information about the **SCRIPTS\_BASEDIR** you are currently in.

A second line prints the logical menu path where you are in, i.e. which menus you have to enter to go to the menu or menupoint. This section is dynamically generated.

**D: DESCRIPTION**

Description text. The output format is justified. If you want to set an empty line, simply enter a **D:** without text.

Example:

```
#
D: The database to be refreshed has been
D: stopped and removed by the Storage Mgmt group.
D:
D: If the "oracle server" or "oracle instance" is
D: "up", contact the Storage Mgmt group.
#
```

**R: RE-RUN ALLOWED?**

The intention of this sections is to either enter *YES* or *NO* or some other text to document if a menu point can be restarted immediately without side effects.

Example:

```
#
R: YES
#
```

**O: OUTPUT-EXAMPLE**

Verbatim output of entered text. Here no text formatting takes place.

Example 1:

```
#
O: PERMISSION OWNER GROUP DIRECTORY
O: drwxr-x--- oracle dba /ora01/oradata/DWHPROD
O: drwxr-x--- oracle dba /ora02/oradata/DWHPROD
O: drwxr-x--- oracle dba /ora03/oradata/DWHPROD
#
```

Example 2 (using **.SS SubSection Title** to create two output sub-sections):

```
#
O: .SS If host acme007 is up:
O: 22:26:53 up 3:51, 1 user, load average: 0.09, 0.15, 0.22
#
O: .SS If host acme007 is not up:
O: ssh-exec-ERROR: remote system 'acme007' is not up, aborting.
#
```

To efficiently get data from a log file, use the **outex(1)** command from within the **vi** session,

using:

```
#
#
~
~
~
: . !outex 2009-09-22_13.10.22__list_db_files.log
```

## L: LIMITATIONS

Document limitations (or bugs) of the recovery script. The output format is justified. If you want to set an empty line, simply enter a **L:** without text.

Example:

```
#
L: The UID (User-ID) and UNAME (User-name)
L: cannot be changed in this tool.
#
```

## N: NOTES

Additional information to the menu point that is not to be documented in the **DESCRIPTION** section. If you want to set an empty line, simply enter a **N:** without text.

Example

```
#
N: Special character input is not
N: allowed (Control-Characters, Umlaut, etc.)
#
```

**F: FILES** List of files. Each text line in this section is printed on a separate line.

Example:

```
#
F: /etc/passwd
F: /etc/nsswitch.conf
F: /var/spool/cron/atjobs/<at-job-file>
#
```

## S: SEE ALSO

Reference to other documentation etc. If the first element of the line is a reference in square brackets (as: [REFER], [1], etc.), it is taken as a reference marker, the following text until the next reference is taken as the reference text.

It is not needed to indent the text as done in the example below for the [AWK] and [CHA] references. The text following a reference can be written as shown for [SSP], the output will be identical for all.

Example:

```
#
S: [AWK] The AWK Programming Language, October 1988,
```

```
S: Aho Alfred V., Weinberger Peter J.,
S: Kernighan Brian W., ISBN 0-201-07981-X
S: [CHA] Clusters, For High Availability,
S: 2nd Edition 2001, Weygant Peter S.,
S: HP-Part No: B3936-90047, ISBN 0-13-089355-2
S: [SSP] Shellscript Programmierung, Sun Service,
S: Revision C21 February 1994, Sun Microsystems Inc.,
S: Sun Part No: 8xx-xxxx-xx
#
```

### T: DURATION

Document the approximate runtime duration of the menu point.

Example:

```
#
T: ~7 minutes
#
```

If using the **scripttitle(3)** command, the duration text given here is displayed as expected duration in braces.

### SPECIAL BEHAVIOR

To speed up documentation writing some special behavior is implemented based on the content written.

- If you start a line with a dash ( - ) the line containing the dash and following text not preceded by a dash are set with an indent of 2 characters. I.e. an unsorted list is created.

Example:

```
#
D: Why go fishing in Canada?
D: - vast area of wild nature where
D: the big fish can be caught.
D: - peaceful lakes and streams invite to
D: relax and recover from daily hassles.
#
```

- /- End an unordered list started with the dash ( - ). This is needed if you like to quit the indented text because the following text does not relate to the list point any more. If the next text you write is assigned to another section, it is not needed to add an /- at the end of the list.

Example:

```
#
D: Why go fishing in Canada?
D: - vast area of non touched nature where
D: the big fish can be caught.
D: - peaceful lakes and streams invite to
D: relax and recover from daily problems.
D: /-
D: Of course there are other places to
D: experience similar landscape as ...
#
```

**preformatted text**

```
<pre>
text
</pre>
```

unformatted output of *text*.

Example:

```
#
D: To efficiently rename many menu points, use:
D: <pre>
D: lsmv > a; vi a; sh ./a; rm a; \
D: scriptheadersync -execute
D: </pre>
D: This loads the 'mv' commands into 'vi' and
#
```

**section tag without text**

If any section tag without text is set, a line break with a line feed occurs. Therefore in the example below, between the whale and the bear part of the text will be a one line gap in the output.

Example:

```
#
D: Beside orcas you can see humpback whales and
D: if you are lucky you will catch a view of a
D: gray whale.
D:
D: It might be the case that you are more
D: interested in bears, those fellows can be
D: seen near ...
#
```

**<br>** force a line break without setting a whole empty line.

Example:

```
#
D: Hint:
D:

D: The 'Be Bear Aware Campaign' is a national,
D: non-profit conservation organization ...
#
```

**<b>text</b>**

set *text* in bold font.

This formatting cannot be used in sections: **F: FILES**, **O: OUTPUT-EXAMPLE**, **F: FILES** and **S: SEE ALSO**.



Example:

```
#
D: Bears are not mean or malicious - they
D: are wild - and all wild animals need
D: their space.
#
```

**<i>text</i>**

set *text* underlined on the terminal and in italic font when producing a HTML output.

This formatting cannot be used in sections: **F: FILES**, **O: OUTPUT-EXAMPLE**, **F: FILES** and **S: SEE ALSO**.

Example:

```
#
D: Blackbears, grizzly bears, polar bears,
D: oh <i>wonderful</i> bears!
#
```

**<u>text</u>**

set *text* underlined on the terminal and in italic font when producing a HTML output.

Basically the **<u>text</u>** is identical to the **<i>text</i>** specification, but to show that the underlining is important to your documenting, this control can be used.

The **<u>text</u>** can be used to "document" that a text is underlined.

This formatting cannot be used in sections: **F: FILES**, **O: OUTPUT-EXAMPLE**, **F: FILES** and **S: SEE ALSO**.

Example:

```
#
D: To see more information about bears,
D: visit the 'Be Bear Aware Campaign':
D:
D: <u>https://bebearaware.org</u>
#
```

## USER DEFINED DOCUMENTATION SECTIONS

It might be the case that the predefined sections as explained above do not fulfill all your needs.

However, it is possible to define additional sections and documentation tags.

To do so you have to list it using the **&:** tag in each recovery script or menu title where you need special sections.

A section definition has the format:

**# &:** *specification* [{ *specification* }]

**specification::= tag:section**

The *tag* is a lowercase letter (a-z), the *section* is the section title. Spaces and tabs are not allowed between the *tag*, the colon ( : ) and the *section* and within the *section* text. If spaces in the *section* title output are needed, they have to be replaced by a % character in the definition.

Example:

```
#
&: e:EMERGENCY%CONTACTS w:WORKAROUNDS
#
#
```

The sections can then be used:

```
#
e: +41 12 345 67 89, 7x24 hour operations
#
w: If you cannot log on to the host dcdbsi20 any more
w: try to connect via the lan console dcdbsi20cons from
w: another server in the farm.
#
```

The new defined sections **EMERGENCY CONTACTS** and **WORKAROUNDS** will be listed after the predefined sections. The output is justified.

**ENVIRONMENT**

-

**EXIT STATUS**

-

**FILES****Recovery Script (=menupoint)**

The recovery script can contain the documentation tags as explained above.

**Menu Title (=menu)**

The title files of menus can contain documentation tags as explained above to add documentation to a menu.

Be aware that the first line of a menu title file contains the title of the menu which is displayed when you invoke the **menu** command in **edrc**. The following lines can contain the documentation tags.

Create/edit/remove the menu title file using the **title** command in **edrc**.

**EXAMPLES**

See documentation of "recovery" script trees distributed with the WA2L/edrc package.

**SEE ALSO**

**contrib(1m)**, **contrib.edrc(1m)**, **edrcintro(1)**, **edrc(1m)**, **edrcenv(1)**, **edrcsetup(1m)**, **outex(1)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

contrib.doc was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

**NAME**

contrib.edrc – list of all contributed commands distributed with WA2L/edrc

**SYNOPSIS**

-

**AVAILABILITY**

WA2L/edrc

**DESCRIPTION**

This man page lists all contributed commands that are part of the core WA2L/edrc package and distributed with it.

If you write own contributed commands do not list it in this man page, due to the fact that your change would be lost after an upgrade of WA2L/edrc. If you send your contributed command to the author of WA2L/edrc your command will be added to this list.

To see where to place manual pages of your own contributed commands, see **contrib(1m)**.

**CONTRIBUTED COMMANDS OF CORE WA2L/edrc**

To see a list of all contributed commands available in your WA2L/edrc installation invoke **help** within **edrc**.

**contrib.apropos(1m)** search the whatis database for strings. See **apropos(1)** for a detail description.

**contrib.bc(1)** An arbitrary precision calculator language that supports arbitrary precision numbers with interactive execution of statements.

There are some similarities in the syntax to the C programming language.

The standard math library is enabled.

See also: **bc(1)**, **contrib.calc(1)**, **contrib.kalc(1)**.

**contrib.calc(1m)** single row calculator using awk. All calculations as possible in **awk(1)** can be performed. Be aware that this command will be denied from startup if *shell* is listed in the **DENY\_LIST**.

See also: **contrib.kalc(1)**, **contrib.bc(1)**.

**contrib.checklist**(1m) print a checklist that can be used to track the disaster recovery progress based on the recovery script tree.

The **checklist** command can also be started from a recovery script.

In this case the environment variable **\$EDRC\_CONTRIB\_CHECKLIST\_ROOTDIR** can be set to produce a checklist for a part of a recovery script tree if needed, as:

```
export EDRC_CONTRIB_CHECKLIST_ROOTDIR=$EDRC_ENTRY_DIR
```

This example produces a checklist starting in the current menu and descending.

In a restricted environment the functionality *contrib.checklist.setrootdir* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the corresponding functionality.

**contrib.clock**(1m) display a running clock banner in military format.

In a restricted environment the functionality *contrib.clock.start* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the corresponding functionality.

**contrib.clonemenu**(1m)

clone (copy) an existing submenu. Pre-requirements for cloning a submenu are, that the source and the clone submenus exist and the clone menu does not have contents (except a title). Therefore ahead of cloning an existing menu it is needed to create a new menu with the **newmenu edrc** command that will be the clone of the existing submenu.

**contrib.cmmon**(1m) monitor of all MC/SG packages on HP-UX systems having MC/ServiceGuard installed. If an interval of 0 is specified a one time output is provided, else the output is continuously.

**contrib.cmswitch**(1m) switch a MC/SG package to a node. Using this interactive command is much faster then entering all commands needed for a cluster package switch from one to another node. You can restart a package if you specify the host where the package is currently running as the target node. Be aware that the **cmswitch** command does not check if it is possible to switch the package to the specified target node, therefore you should check the possible package switch positions first using the **cmmon** contributed command.

**contrib.cmviewcl**(1m) MC/ServiceGuard state. This is the presentation of the standard **cmviewcl**(1) command available on HP-UX with installed MC/ServiceGuard to **edrc**.

**contrib.countdown(1m)**

countdown control (a stopwatch). The values are stored persistently, therefore after a system reboot the countdown figures are still valid.

In a restricted environment the functionality *contrib.countdown.init*, *contrib.countdown.pause*, *contrib.countdown.resume*, *contrib.countdown.end* and *contrib.countdown.start* can be added to the **DENY\_LIST** of the **edrc.cfg(4)** file to deny the corresponding functionality.

The fore- and background colors can optionally be defined in the global config file **etc/contrib.edrc.countdown.cfg** with the settings **COUNTDOWN\_FOREGROUND\_COLOR=***color* and **COUNTDOWN\_BACKGROUND\_COLOR=***color* where the *color* values are numbers between 0 and 7 (black= 0, red= 1, green= 2, yellow= 3, blue= 4, magenta= 5, cyan= 6, white= 7).

To set the character used to display the big counter (that is normally composed out of "#" characters, use the **COUNTDOWN\_COUNTER\_CHARACTER="***character***"** setting to define a single character to be used instead. You can also use a block character, as: "â" to produce a more prominent counter effect.

The countdown state can also be viewed thru **wa2ledrc\_report(1)**.

**contrib.dirwatch(1m)**

watch and summarize restore progress. With this command directory sizes are monitored and the average size difference over time is calculated. Therefore this command can be used to monitor restore, removal or file copy progress and do an estimation when it would be completed.

**contrib.doc(1m)**

display recovery script documentation. All menu points can contain a documentation that is displayed using this command. If no documentation tags are found in a menu point, only the dynamic documentation header is displayed.

**contrib.drstat(1m)**

LTO library drive status. This command is useful to display the tape drive states of a tape library on HP-UX systems where HP DataProtector (former OmniBack) is installed.

**contrib.edrcenv(1m)**

print the official WA2L/edrc environment variables as the **edrcenv** command. See also **edrcenv(1m)**.

**contrib.edrcupgrade(1m)**

an easy to use interface to efficiently download WA2L/edrc patches from **sourceforge.net**.

In addition the downloaded patches can be deployed and installed. If a direct internet connection is not permitted from the system the connection via a http proxy is also supported. Furthermore the patches could also be provided on a internal web server or locally on the filesystem from where the patch installation should be performed.

In a restricted environment the functionality *contrib.edrcupgrade.start*, *contrib.edrcupgrade.download*, *contrib.edrcupgrade.install*, *contrib.edrcupgrade.setagent*, *contrib.edrcupgrade.setdir*, *contrib.edrcupgrade.useproxy*, *contrib.edrcupgrade.setproxy*, *contrib.edrcupgrade.setproxyauth*, *contrib.edrcupgrade.sethttpauth* and *contrib.edrcupgrade.settargets* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the corresponding functionality.

**contrib.find**(1m) find text in recovery scripts header and documentation and print a list of matched menu points. The text to find is a non case sensitive regular expression. See also: **contrib.search**(1m).

**contrib.kalc**(1) A programmable scientific calculator, using RPN (Reverse Polish Notation). It includes over 200 functions and a built-in help system. **kalc** works with real numbers, complex numbers and also integers in other numeric bases. Its has a complete programming language, with control-flow structures such as if and while. It has "unlimited" memory for you to store objects.

See also: **bc**(1), **contrib.bc**(1), **contrib.calc**(1m).

**contrib.leo**(1m) execute dictionary queries to **http://dict.leo.org**. The input defaults and options can optionally be defined in the global config file **etc/contrib.edrc.leo.cfg** which has the identical format as the file **var/contrib/edrc/leo.<session name>** where the inputs are persistently saved in an **edrc** session.

To change the translation language or **leo** options, enter **!** in the **'term to translate'** prompt.

See **leo**(1) for possible language and options settings.

In a restricted environment the functionality *contrib.leo.setlang*, *contrib.leo.setproxy* and *contrib.leo.setopts* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the corresponding functionality.

**contrib.license**(1m) print the WA2L/edrc license terms.

**contrib.locations**(1m) print a list of corporate locations with local time.

**contrib.logs**(1m) view any session log file of **edrc**. This is an enhanced log viewer compared to the internal **log** command of **edrc** which displays the log file of the last script run. It is possible to select the logfiles by date or session and to "scroll" efficiently thru a sequence of logs.

In a restricted environment the functionality *contrib.logs.view* should be added to the **DENY\_LIST** of the **edrc.cfg**(4) file.

**contrib.logtail(1m)** give a selection of common logfiles to be displayed with **tail -f** *<selected logfile>* . This is the presentation of the **logtail(1)** command to **edrc**.

**contrib.logview(1m)** give a selection of common logfiles to be displayed with **view** *<selected logfile>* . This is the presentation of the **logview(1)** command to **edrc**.

**contrib.logviewer(1m)** create a browseable LogViewer HTML archive containing all log files between two selected dates. The generated **.zip** file should be encrypted due to the fact that there might be a chance that a logfile could contain passwords.

In **\$EDRC\_SCRIPTS\_BASEDIR/logviewer.fl** a list of extra log files also to be included to the LogViewer archive can be defined. Relative path names are relative to the WA2L/edrc installation.

In a restricted environment the functionality *contrib.logviewer.sendpassword*, *contrib.logviewer.setaddress*, *contrib.logviewer.setdates*, *contrib.logviewer.setdir*, *contrib.logviewer.setemptyxlogs*, *contrib.logviewer.setmode* and *contrib.logviewer.setpassword* can be added to the **DENY\_LIST** of the **edrc.cfg(4)** file to deny the related functionality in the command.

**contrib.motd(1m)** print system wide message of the day (*/etc/motd*).

**contrib.newscripttree(1m)** create a new script tree based on a template.

**contrib.notepad(1m)** a notepad to quickly save ad hoc notes locally.

If the setting **TIMESTAMPS=ON** is set within the notepad file, automatically a timestamp in the military format is added when opening the file for editing.

In a restricted environment the functionality *contrib.notepad.edit*, *contrib.notepad.view*, *contrib.notepad.display*, *contrib.notepad.more* and *contrib.countdown.grep* can be added to the **DENY\_LIST** of the **edrc.cfg(4)** file to deny the corresponding functionality.

The notepad has the name **notepad.<edrc\_config>.txt** and is saved to the **\$EDRC\_CONTRIB\_VARDIR** by default. The save directory can optionally be defined in the global config file **etc/contrib.edrc.notepad.cfg** with the settings **NOTEPAD\_NOTESDIR=directory** .

**contrib.oerr(1m)** resolve Oracle error numbers. This is the presentation of the **oerr(1)** command available on systems where Oracle is installed to **edrc**.

**contrib.omnimon(1m)** OmniBack Session Monitor. With this commands the status of HP DataProtector (former OmniBack) can be continuously displayed on cell servers where



DataProtector is installed.

**contrib.omnistat**(1m) OmniBack Status (running sessions).

**contrib.pwsafe**(1m) interactive password safe that allows also to protect the password data file with a master password if needed.

To change the master password, invoke: **pwsafe -> edit -> change some content -> save**

The password data file is saved in the **\$EDRC\_SCRIPTS\_BASEDIR/pwsafe.dat** file which will be newly created from a template when **pwsafe** is initially called.

The only compulsory field in the password data file is **HANDLE**; which has to be the leftmost one.

If the direct query functionality **pwsafe user@system** is used the **PASSWORD**;, **USER**;, and **SYSTEM\_OR\_APPLICATION**;; fields must also exist.

All other columns are optional and can be changed.

In a restricted environment the functionality *contrib.pwsafe.edit* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the functionality to edit the password data file and *contrib.pwsafe.directquery* to deny direct **pwsafe user@system** query within recovery scripts.

Because the **pwsafe** saves the data as semicolon (;) separated file that allows tabs and spaces to format the file, a semicolon, space and tabulator cannot be entered plain in the **PASSWORD** field. Therefore if your password contains a semicolon, replace it with **[:SEMICOLON:]**, if it contains a space, replace it with **[:SPACE:]** and replace a tabulator with **[:TAB:]**.

So if your password is **'my;secret '** the password has to be defined as **my[:SEMICOLON:]secret[:SPACE:]** in the **PASSWORD** field of the **pwsafe**.

The replacement strings **[:SEMICOLON:]**, **[:SPACE:]** and **[:TAB:]** can be changed in the optional configuration file **etc/contrib.edrc.pwsafe.cfg** using the **SEMICOLON=string**, **SPACE=string** and **TAB=string** settings.

The encryption mode default is **AES** which uses **ccrypt**(1).

The encryption mode can be defined in the optional configuration file **etc/contrib.edrc.pwsafe.cfg** using the setting **CRYPT\_MODE=mode** (example: **CRYPT\_MODE=AES256**).

Possible encryption modes are:

**AES** uses **ccrypt**(1) by Peter Selinger.

It is based on the Rijndael block cipher, a version of which is also used in the Advanced Encryption Standard (AES, see <http://www.nist.gov/aes>). This cipher is believed to provide very strong security.

**AES256**

uses **openssl(1)**

The openssl program is a command line tool for using the various cryptography functions of OpenSSL's crypto library from the shell. This cipher is believed to provide very strong security.

**ZIP** uses **zip(1)**

Where security is truly important, use strong encryption such as **AES** or **AES256** instead of the relatively weak standard encryption provided by zip-file utilities.

**ENIGMA**

uses **crypt(1)**

Where security is truly important, use strong encryption such as **AES** or **AES256** instead of the relatively weak encryption provided by the enigma encryption of **crypt(1)**.

**ENIGMA** is provided to handle password safe files produced in early versions of WA2L/edrc but should not be used for new setups.

**NO** uses no encryption even if you answer the dialog to encrypt the file with **y** (yes).

This mode could be used for intermediate solutions where none of the commands needed by the other modes are working on a system.

In general a **pwsafe.nolog** file should exist in the **edrc/contrib/edrc** directory to avoid output logging for this command. This is especially important if you decide to protect the password data file with a master password because else the clear text passwords will be seen in the corresponding log files.

The **pwsafe(3)** command that can be called from outside of **edrc** ( using the command: **edrc/lib/pwsafe [ -c config\_file ] user@system** ) allows also to access the password safe of a recovery script tree.

In this case the master password has to be provided thru the environment variable **\$PWSAFE\_MASTERPASSWORD**.

Of course one using this command should be careful about how to handle the master password.

See **pwsafe(3)** for more information.

**contrib.pid(1m)**

UNIX process list. Select processes from the system. See **pid(1)** for a detail description.

**contrib.ping(1m)**

send ICMP ECHO\_REQUEST packets to network hosts. This interactive command guides thru a more advanced usage of **ping(8)**.

**contrib.pstree**(1m) UNIX process tree. The processes of an optionally specified root process is displayed in a tree form.

**contrib.sane**(1m) reset the tty issuing **stty sane** internally.

**contrib.scriptdocbook**(1m)

create a document book in WA2L/DocBrowser of the current recovery script tree. This command travels thru the whole script tree and generates a HTML page for each menu point. The information in the HTML page is as the **doc** contributed command is called.

In a restricted environment the functionality *contrib.scriptdocbook.checklist*, *contrib.scriptdocbook.setdir*, *contrib.scriptdocbook.setmode*, *contrib.scriptdocbook.setpassword*, *contrib.scriptdocbook.sendpassword* and *contrib.scriptdocbook.setaddress* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the related functionality in the command.

A BIBLIOGRAPHY entry is added as the very end of the menu, if the file **scriptdocbook.bib** is created in the **\$EDRC\_SCRIPTS\_BASEDIR** directory. The entries must have the following format:

```
[EDRC]
TITLE= EDRC
SUBTITLE= Enterprise Disaster Recovery Console
VERSION= 1.5.32
DATE= 30.01.2010
COMPANY= ACME Switzerland
AUTHOR= Walther Christian
ISBN= ISBN 0-330-35169-9
FILE= File: edrc_manpages-1.5.32.pdf
PATH= doc/edrc_manpages-1.*.pdf
URL= http://www.acme.ch/doc/edrc.html
```

The compulsory entry is *[reference]*, all other entries are optional.

If the *[----]* reference is used, a horizontal line is inserted. For this special reference no other fields should be set.

The **PATH=***path/filename* defines the path- and filename of the file referenced in the **FILE** entry.

The *path/filename* can be specified as follows:

*/path/filename*  
absolute file location on the system.

*~user/path/filename*  
file located in a **\$HOME** directory of a *user*.

*./path/filename*

file located in the **\$EDRC\_SCRIPTS\_BASEDIR**

*path/filename*

file located relatively to the WA2L/edrc package installation directory.

If the file in **PATH** exists, it is copied to the ScriptDocBook and the entry in **FILE** is a hyperlink pointing to that included file.

If **TODAY** is specified in *date* of **DATE=date**, the current date will be printed. This might be useful when including configuration files into the ScriptDocBook.

When **EDRCREVISION** is specified in *version* of **VERSION=version** the current WA2L/edrc release is printed.

The input defaults can optionally be defined in the global config file **etc/contrib.edrc.scriptdocbook.cfg** and/or the local "recovery" script tree config file **\$EDRC\_SCRIPTS\_BASEDIR/scriptdocbook.cfg** which has the identical format as the file **var/contrib/edrc/scriptdocbook.<session name>** where the inputs are persistently saved in an **edrc** session.

#### **contrib.scriptinstall(1m)**

install a script tree package created with the **contrib.scriptpack(1m)** contributed command into the current script tree.

The current script tree contents is removed (and backed up to the **\$EDRC\_SCRIPTS\_BASEDIR/.sav/scriptinstall/<timestamp>/** directory) before the installation of the script tree package file.

In a restricted environment the functionality *contrib.scriptinstall.install*, *contrib.scriptinstall.list* and *contrib.scriptinstall.patch* can be added to the **DENY\_LIST** of the **edrc.cfg(4)** file to deny the related functionality of the **scriptinstall** contributed command.

#### **contrib.scriptpack(1m)**

pack the current recovery script tree to an archive package (= \*.cpio.gz file).

It is supported to stripe out the passwords stored in **\_env** files and to exclude an existing **pwsafe.dat** file from the script package based on dialog input. Furthermore it can be selected if the recovery script tree package should be saved into a directory or if it should be mailed to a list of recipients.

The generated package can be shared among other system administrators, can be used as a backup snapshot of a recovery tree at a certain point in time (e.g. as proof after a disaster recovery (test)) or to transport a script tree from one environment to another (if a direct **distribute** is not possible or the code transport between environments needs to be carried out by other personnel due to segregation of duty requirements).

The generated package can be installed most efficiently on the target system using the **contrib.scriptinstall(1m)** contributed command.

To easily list/cat the contents of the generated package file, the **llcomp**(1), **lscomp**(1) and **catcomp**(1) can be used.

In a restricted environment the functionality *contrib.scriptpack.cpconfig*, *contrib.scriptpack.cpfiles*, *contrib.scriptpack.setdir*, *contrib.scriptpack.passwdstrip*, *contrib.scriptpack.pwsaferm*, *contrib.scriptpack.setmode*, *contrib.scriptpack.setaddress*, *contrib.scriptpack.sendpassword* and *contrib.scriptpack.setpassword* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the related functionality of the **scriptpack** contributed command.

The input defaults can optionally be defined in the config file **etc/contrib.edrc.scriptpack.cfg** which has the identical format as the file **var/contrib/edrc/scriptpack.<session name>** where the inputs are persistently saved in an **edrc** session.

In **\$EDRC\_SCRIPTS\_BASEDIR/scriptpack.fl** a list of other files (one per line) and directories also to be saved to the script package can be defined. Relative path names are relative to the WA2L/edrc installation.

The (include) list can contain wildcards as known from the **ls** and **find** commands. To test an entry run the '**find entry**' command.

To exclude files from the resulting list of (included) files, list it in the **[EXCLUDE]** section, whereas the list entries are treated as regular expressions.

All files before a **[EXCLUDE]** section are treated as includes. If the includes shall be defined after the **[EXCLUDE]** section, an **[INCLUDE]** section has to be defined, else the **[INCLUDE]** keyword is optional.

The files listed in **scriptpack.fl** are packed to the **\$EDRC\_SCRIPTS\_BASEDIR/.sav/scriptpack/file/** directory and the WA2L/edrc configuration files (if chosen in the **scriptpack** dialog) to **\$EDRC\_SCRIPTS\_BASEDIR/.sav/scriptpack/conf/** in the script tree package file.

#### **contrib.scriptsequence**(1m)

globally control the **scriptsequence**(3) command used in **\_env** files or recovery scripts during a session without changing the related file.

When the (m)ode is set to **dynamic** (=default) the action as defined in the **scriptsequence -a action** command line option is used as specified, therefore if **scriptsequence** is used with other actions in other menus, the related action is used.

If the (m)ode is set to **query**, **inform** or **cancel** the setting specified in the contributed command overrides the command line options of **scriptsequence** for the whole session.

When it is chosen to (d)isable and the script sequence checking, the auditing of the direct ascending menu point starts is set on hold as long as it is chosen to (e)nable the checking again (or the session is newly started).

In a restricted environment the functionality *contrib.scriptsequence.print*, *contrib.scriptsequence.disable*, *contrib.scriptsequence.enable*, *contrib.scriptsequence.init* and *contrib.scriptsequence.setmode* can be added to the **DENY\_LIST**

of the **edrc.cfg**(4) file to deny the related functionality of the **scriptsequence** contributed command.

**contrib.scripttree**(1m) print recovery script menu tree. A tree of all scripts and menus in a "Recovery" script tree configuration is printed. If you are looking for a certain menupoint in a big recovery script tree, this command might help.

**contrib.scriptstats**(1m)  
print recovery script menu statistics.

**contrib.search**(1m) search text in recovery scripts. The text to search is a case sensitive regular expression. See also: **contrib.find**(1m).

In a restricted environment the functionality *contrib.search.all*, *contrib.search.code*, *contrib.search.doc*, *contrib.search.file*, and *contrib.search.header*, can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the related functionality.

**contrib.sessions**(1m) display currently running **edrc**(1m) sessions on the local host. It is also possible to end possibly existing inactive **edrc** sessions.

In a restricted environment the functionality *contrib.sessions.end* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the functionality to end inactive sessions.

**contrib.tail**(1m) monitor the growth of a local or remote file using **tail -f file** internally.

To monitor a local file, specify */var/log/mylogfile.log* at the '**file to monitor:**' prompt.

To monitor a remote file specify *hostname:/var/log/mylogfile.log* or *remoteuser@hostname:/var/log/mylogfile.log* at the '**file to monitor:**' prompt.

In a restricted environment the functionality *contrib.tail.remote* can be added to the **DENY\_LIST** of the **edrc.cfg**(4) file to deny the functionality to **tail -f** a remote file.

**contrib.whatis**(1m) search the **whatis** database for strings. See **whatis**(1) for a detail description.

**contrib.whereami**(1m) server environment where logged on. Print the server environment where you are currently logged on. This command helps if you are unsure if you are really logged on to the TEST environment, for instance.

**ENVIRONMENT**

See **contrib(1m)** and **edrc(1m)** section **ENVIRONMENT**.

**EXIT STATUS**

-

**FILES**

See **contrib(1m)** and **contrib.<contributed\_command>(1m)**.

**EXAMPLES**

-

**SEE ALSO**

**edrcintro(1)**, **bc(1)**, **edrc(1m)**, **edrc.cfg(4)**, **edrcenv(1)**, **contrib(1m)**, **contrib.apropos(1m)**, **contrib.bc(1)**, **contrib.calc(1m)**, **contrib.checklist(1m)**, **contrib.clock(1m)**, **contrib.clonemenu(1m)**, **contrib.cmmon(1m)**, **contrib.cmswitch(1m)**, **contrib.cmviewcl(1m)**, **contrib.countdown(1m)**, **contrib.dirwatch(1m)**, **contrib.doc(1m)**, **contrib.drstat(1m)**, **contrib.edrcupgrade(1m)**, **contrib.find(1m)**, **contrib.kalc(1)**, **contrib.leo(1m)**, **contrib.license(1m)**, **contrib.locations(1m)**, **contrib.logs(1m)**, **contrib.logtail(1m)**, **contrib.logview(1m)**, **contrib.logviewer(1m)**, **contrib.motd(1m)**, **contrib.newscript-tree(1m)**, **contrib.oerr(1m)**, **contrib.omnimon(1m)**, **contrib.omnistat(1m)**, **contrib.pwsafe(1m)**, **contrib.pid(1m)**, **contrib.ping(1m)**, **contrib.pstree(1m)**, **contrib.sane(1m)**, **contrib.scriptdocbook(1m)**, **contrib.scriptinstall(1m)**, **contrib.scriptpack(1m)**, **contrib.edrc.scriptpack.cfg(4)**, **contrib.scripttree(1m)**, **contrib.scriptsequence(1m)**, **contrib.scriptstats(1m)**, **contrib.search(1m)**, **contrib.sessions(1m)**, **contrib.tail(1m)**, **contrib.whatis(1m)**, **contrib.whereami(1m)**, **contrib.<contributed\_command>(1m)**, **is\_permitted(3)**, **pwsafe(3)**

**NOTES**

-

**BUGS**

-

**AUTHOR**

contrib.edrc was developed by Christian Walther. Send suggestions and bug reports to [wa2l@users.sourceforge.net](mailto:wa2l@users.sourceforge.net).

**COPYRIGHT**

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.



**NAME**

**cpanm** – get, unpack build and install modules from CPAN

**SYNOPSIS**

```

cpanm Test::More # install Test::More
cpanm MIYAGAWA/Plack-0.99_05.tar.gz # full distribution path
cpanm http://example.org/LDS/CGI.pm-3.20.tar.gz # install from URL
cpanm ~/dists/MyCompany-Enterprise-1.00.tar.gz # install from a local file
cpanm --interactive Task::Kensho # Configure interactively
cpanm . # install from local directory
cpanm --installdeps . # install all the deps for the
cpanm -L extlib Plack # install Plack and all non-co.
cpanm --mirror http://cpan.cpantesters.org/ DBI # use the fast-syncing mirror
cpanm --from https://cpan.metacpan.org/ Plack # use only the HTTPS mirror

```

**COMMANDS**

(arguments)

Command line arguments can be either a module name, distribution file, local file path, HTTP URL or git repository URL. Following commands will all work as you expect.

```

cpanm Plack
cpanm Plack/Request.pm
cpanm MIYAGAWA/Plack-1.0000.tar.gz
cpanm /path/to/Plack-1.0000.tar.gz
cpanm http://cpan.metacpan.org/authors/id/M/MI/MIYAGAWA/Plack-0.9990.tar.g
cpanm git://github.com/plack/Plack.git

```

Additionally, you can use the notation using `~` and `@` to specify version for a given module. `~` specifies the version requirement in the CPAN::Meta::Spec format, while `@` pins the exact version, and is a shortcut for `~"== VERSION"`.

```

cpanm Plack~1.0000 # 1.0000 or later
cpanm Plack~">= 1.0000, < 2.0000" # latest of 1.xxxx
cpanm Plack@0.9990 # specific version. same as Plack~"== 0

```

The version query including specific version or range will be sent to MetaCPAN to search for previous releases. The query will search for BackPAN archives by default, unless you specify `--dev` option, in which case, archived versions will be filtered out.

For a git repository, you can specify a branch, tag, or commit SHA to build. The default is master

```

cpanm git://github.com/plack/Plack.git@1.0000 # tag
cpanm git://github.com/plack/Plack.git@devel # branch

```

**-i, --install**

Installs the modules. This is a default behavior and this is just a compatibility option to make it work like `cpan` or `cpanp`.

**--self-upgrade**

Upgrades itself. It's just an alias for:

```
cpanm App::cpanminus
```

**--info**

Displays the distribution information in `AUTHOR/Dist-Name-ver.tar.gz` format in the standard out.

**--installdeps**

Installs the dependencies of the target distribution but won't build itself. Handy if you want to try the application from a version controlled repository such as git.

- ```
cpanm --installdeps .
```
- `--look`
Download and unpack the distribution and then open the directory with your shell. Handy to poke around the source code or do manual testing.
 - `-h, --help`
Displays the help message.
 - `-V, --version`
Displays the version number.

OPTIONS

You can specify the default options in `PERL_CPANM_OPT` environment variable.

- `-f, --force`
Force install modules even when testing failed.
- `-n, --notest`
Skip the testing of modules. Use this only when you just want to save time for installing hundreds of distributions to the same perl and architecture you've already tested to make sure it builds fine.

Defaults to false, and you can say `--no-notest` to override when it is set in the default options in `PERL_CPANM_OPT`.
- `--test-only`
Run the tests only, and do not install the specified module or distributions. Handy if you want to verify the new (or even old) releases pass its unit tests without installing the module.

Note that if you specify this option with a module or distribution that has dependencies, these dependencies will be installed if you don't currently have them.
- `-S, --sudo`
Switch to the root user with `sudo` when installing modules. Use this if you want to install modules to the system perl include path.

Defaults to false, and you can say `--no-sudo` to override when it is set in the default options in `PERL_CPANM_OPT`.
- `-v, --verbose`
Makes the output verbose. It also enables the interactive configuration. (See `--interactive`)
- `-q, --quiet`
Makes the output even more quiet than the default. It only shows the successful/failed dependencies to the output.
- `-l, --local-lib`
Sets the `local::lib` compatible path to install modules to. You don't need to set this if you already configure the shell environment variables using `local::lib`, but this can be used to override that as well.
- `-L, --local-lib-contained`
Same with `--local-lib` but with `--self-contained` set. All non-core dependencies will be installed even if they're already installed.

For instance,

```
cpanm -L extlib Plack
```

would install Plack and all of its non-core dependencies into the directory `extlib`, which can be loaded from your application with:

```
use local::lib '/path/to/extlib';
```

Note that this option does **NOT** reliably work with perl installations supplied by operating system vendors that strips standard modules from perl, such as RHEL, Fedora and CentOS, **UNLESS** you also install packages supplying all the modules that have been stripped. For these systems you will

probably want to install the `perl-core` meta-package which does just that.

--self-contained

When examining the dependencies, assume no non-core modules are installed on the system. Handy if you want to bundle application dependencies in one directory so you can distribute to other machines.

--exclude-vendor

Don't include modules installed under the 'vendor' paths when searching for core modules when the `--self-contained` flag is in effect. This restores the behaviour from before version 1.7023

--mirror

Specifies the base URL for the CPAN mirror to use, such as `http://cpan.cpantesters.org/` (you can omit the trailing slash). You can specify multiple mirror URLs by repeating the command line option.

You can use a local directory that has a CPAN mirror structure (created by tools such as OrePAN or Pinto) by using a special URL scheme `file:///`. If the given URL begins with '/' (without any scheme), it is considered as a file scheme as well.

```
cpanm --mirror file:///path/to/mirror
cpanm --mirror ~/minicpan          # Because shell expands ~ to /home/user
```

Defaults to `http://www.cpan.org/`.

--mirror-only

Download the mirror's `02packages.details.txt.gz` index file instead of querying the CPAN Meta DB. This will also effectively opt out sending your local perl versions to backend database servers such as CPAN Meta DB and MetaCPAN.

Select this option if you are using a local mirror of CPAN, such as minicpan when you're offline, or your own CPAN index (a.k.a darkpan).

--from, -M

```
cpanm -M https://cpan.metacpan.org/
cpanm --from https://cpan.metacpan.org/
```

Use the given mirror URL and its index as the *only* source to search and download modules from.

It works similar to `--mirror` and `--mirror-only` combined, with a small difference: unlike `--mirror` which *appends* the URL to the list of mirrors, `--from` (or `-M` for short) uses the specified URL as its *only* source to download index and modules from. This makes the option always override the default mirror, which might have been set via global options such as the one set by `PERL_CPANM_OPT` environment variable.

Tip: It might be useful if you name these options with your shell aliases, like:

```
alias minicpanm='cpanm --from ~/minicpan'
alias darkpan='cpanm --from http://mycompany.example.com/DPAN'
```

--mirror-index

EXPERIMENTAL: Specifies the file path to `02packages.details.txt` for module search index.

--cpanmetadb

EXPERIMENTAL: Specifies an alternate URI for CPAN MetaDB index lookups.

--metacpan

Prefers MetaCPAN API over CPAN MetaDB.

--cpanfile

EXPERIMENTAL: Specified an alternate path for `cpanfile` to search for, when `--installdeps` command is in use. Defaults to `cpanfile`.

--prompt

Prompts when a test fails so that you can skip, force install, retry or look in the shell to see what's going wrong. It also prompts when one of the dependency failed if you want to proceed the installation.

Defaults to false, and you can say `--no-prompt` to override if it's set in the default options in `PERL_CPANM_OPT`.

--dev

EXPERIMENTAL: search for a newer developer release as well. Defaults to false.

--reinstall

`cpanm`, when given a module name in the command line (i.e. `cpanm Plack`), checks the locally installed version first and skips if it is already installed. This option makes it skip the check, so:

```
cpanm --reinstall Plack
```

would reinstall `Plack` even if your locally installed version is latest, or even newer (which would happen if you install a developer release from version control repositories).

Defaults to false.

--interactive

Makes the configuration (such as `Makefile.PL` and `Build.PL`) interactive, so you can answer questions in the distribution that requires custom configuration or `Task::distributions`.

Defaults to false, and you can say `--no-interactive` to override when it's set in the default options in `PERL_CPANM_OPT`.

--pp, --pureperl

Prefer Pure perl build of modules by setting `PUREPERL_ONLY=1` for `MakeMaker` and `--pureperl-only` for `Build.PL` based distributions. Note that not all of the CPAN modules support this convention yet.

--with-recommends, --with-suggests

EXPERIMENTAL: Installs dependencies declared as `recommends` and `suggests` respectively, per META spec. When these dependencies fail to install, `cpanm` continues the installation, since they're just recommendation/suggestion.

Enabling this could potentially make a circular dependency for a few modules on CPAN, when `recommends` adds a module that `recommends` back the module in return.

There's also `--without-recommend` and `--without-suggests` to override the default decision made earlier in `PERL_CPANM_OPT`.

Defaults to false for both.

--with-develop

EXPERIMENTAL: Installs develop phase dependencies in META files or `cpanfile` when used with `--installdeps`. Defaults to false.

--with-configure

EXPERIMENTAL: Installs configure phase dependencies in `cpanfile` when used with `--installdeps`. Defaults to false.

--with-feature, --without-feature, --with-all-features

EXPERIMENTAL: Specifies the feature to enable, if a module supports optional features per META spec 2.0.

```
cpanm --with-feature=opt_csv Spreadsheet::Read
```

the features can also be interactively chosen when `--interactive` option is enabled.

`--with-all-features` enables all the optional features, and `--without-feature` can select a feature to disable.

`--configure-timeout`, `--build-timeout`, `--test-timeout`

Specify the timeout length (in seconds) to wait for the configure, build and test process. Current default values are: 60 for configure, 3600 for build and 1800 for test.

`--configure-args`, `--build-args`, `--test-args`, `--install-args`

EXPERIMENTAL: Pass arguments for configure/build/test/install commands respectively, for a given module to install.

```
cpanm DBD:mysql --configure-args="--cflags=... --libs=..."
```

The argument is only enabled for the module passed as a command line argument, not dependencies.

`--scandeps`

DEPRECATED: Scans the dependencies of given modules and output the tree in a text format. (See `--format` below for more options)

Because this command doesn't actually install any distributions, it will be useful that by typing:

```
cpanm --scandeps Catalyst::Runtime
```

you can make sure what modules will be installed.

This command takes into account which modules you already have installed in your system. If you want to see what modules will be installed against a vanilla perl installation, you might want to combine it with `-L` option.

`--format`

DEPRECATED: Determines what format to display the scanned dependency tree. Available options are `tree`, `json`, `yaml` and `dists`.

`tree` Displays the tree in a plain text format. This is the default value.

`json`, `yaml`

Outputs the tree in a JSON or YAML format. JSON and YAML modules need to be installed respectively. The output tree is represented as a recursive tuple of:

```
[ distribution, dependencies ]
```

and the container is an array containing the root elements. Note that there may be multiple root nodes, since you can give multiple modules to the `--scandeps` command.

`dists` `dists` is a special output format, where it prints the distribution filename in the *depth first order* after the dependency resolution, like:

```
GAAS/MIME-Base64-3.13.tar.gz
GAAS/URI-1.58.tar.gz
PETDANCE/HTML-Tagset-3.20.tar.gz
GAAS/HTML-Parser-3.68.tar.gz
GAAS/libwww-perl-5.837.tar.gz
```

which means you can install these distributions in this order without extra dependencies. When combined with `-L` option, it will be useful to replay installations on other machines.

`--save-dists`

Specifies the optional directory path to copy downloaded tarballs in the CPAN mirror compatible directory structure i.e. `authors/id/A/AU/AUTHORS/Foo-Bar-version.tar.gz`

If the distro tarball did not come from CPAN, for example from a local file or from GitHub, then it will be saved under `vendor/Foo-Bar-version.tar.gz`.

`--uninst-shadows`

Uninstalls the shadow files of the distribution that you're installing. This eliminates the confusion if you're trying to install core (dual-life) modules from CPAN against perl 5.10 or older, or modules that used to be XS-based but switched to pure perl at some version.

If you run `cpanm` as root and use `INSTALL_BASE` or equivalent to specify custom installation path, you **SHOULD** disable this option so you won't accidentally uninstall dual-life modules from the core include path.

Defaults to true if your perl version is smaller than 5.12, and you can disable that with `--no-uninst-shadows`.

NOTE: Since version 1.3000 this flag is turned off by default for perl newer than 5.12, since with 5.12 `@INC` contains `site_perl` directory *before* the perl core library path, and uninstalling shadows is not necessary anymore and does more harm by deleting files from the core library path.

`--uninstall, -U`

Uninstalls a module from the library path. It finds a packlist for given modules, and removes all the files included in the same distribution.

If you enable `local::lib`, it only removes files from the `local::lib` directory.

If you try to uninstall a module in `perl` directory (i.e. core module), an error will be thrown.

A dialog will be prompted to confirm the files to be deleted. If you pass `-f` option as well, the dialog will be skipped and uninstallation will be forced.

`--cascade-search`

EXPERIMENTAL: Specifies whether to cascade search when you specify multiple mirrors and a mirror doesn't have a module or has a lower version of the module than requested. Defaults to false.

`--skip-installed`

Specifies whether a module given in the command line is skipped if its latest version is already installed. Defaults to true.

NOTE: The `PERL5LIB` environment variable have to be correctly set for this to work with modules installed using `local::lib`, unless you always use the `-l` option.

`--skip-satisfied`

EXPERIMENTAL: Specifies whether a module (and version) given in the command line is skipped if it's already installed.

If you run:

```
cpanm --skip-satisfied CGI DBI~1.2
```

`cpanm` won't install them if you already have `CGI` (for whatever versions) or have `DBI` with version higher than 1.2. It is similar to `--skip-installed` but while `--skip-installed` checks if the *latest* version of CPAN is installed, `--skip-satisfied` checks if a requested version (or not, which means any version) is installed.

Defaults to false.

`--verify`

Verify the integrity of distribution files retrieved from CPAN using `CHECKSUMS` file, and `SIGNATURES` file (if found in the distribution). Defaults to false.

Using this option does not verify the integrity of the `CHECKSUMS` file, and it's unsafe to rely on this option if you're using a CPAN mirror that you do not trust.

`--report-perl-version`

Whether it reports the locally installed perl version to the various web server as part of User-Agent. Defaults to true unless CI related environment variables such as `TRAVIS`, `CI` or `AUTOMATED_TESTING` is enabled. You can disable it by using `--no-report-perl-version`.

`--auto-cleanup`

Specifies the number of days in which `cpanm`'s work directories expire. Defaults to 7, which means old work directories will be cleaned up in one week.

You can set the value to 0 to make `cpan` never cleanup those directories.

--man-pages

Generates man pages for executables (man1) and libraries (man3).

Defaults to true (man pages generated) unless `-L|--local-lib-contained` option is supplied in which case it's set to false. You can disable it with `--no-man-pages`.

--lwp

Uses LWP module to download stuff over HTTP. Defaults to true, and you can say `--no-lwp` to disable using LWP, when you want to upgrade LWP from CPAN on some broken perl systems.

--wget

Uses GNU Wget (if available) to download stuff. Defaults to true, and you can say `--no-wget` to disable using Wget (versions of Wget older than 1.9 don't support the `--retry-connrefused` option used by cpanm).

--curl

Uses cURL (if available) to download stuff. Defaults to true, and you can say `--no-curl` to disable using cURL.

Normally with `--lwp`, `--wget` and `--curl` options set to true (which is the default) cpanm tries LWP, Wget, cURL and HTTP::Tiny (in that order) and uses the first one available.

ENVIRONMENT VARIABLES**PERL_CPANM_HOME**

The directory cpanm should use to store downloads and build and test modules. Defaults to the `.cpanm` directory in your user's home directory.

PERL_CPANM_OPT

If set, adds a set of default options to every cpanm command. These options come first, and so are overridden by command-line options.

SEE ALSO

`App::cpanminus`

COPYRIGHT

Copyright 2010– Tatsuhiko Miyagawa.

AUTHOR

Tatsuhiko Miyagawa

NAME

cpio – copy files to and from archives

SYNOPSIS

```
cpio {-o|--create} [-0acvABLV] [-C bytes] [-H format] [-M message] [-O [[user@]host:]archive] [-F
[[user@]host:]archive] [--file=[[user@]host:]archive] [--format=format] [--message=message] [--null]
[--reset-access-time] [--verbose] [--dot] [--append] [--block-size=blocks] [--dereference] [--io-
size=bytes] [--quiet] [--force-local] [--rsh-command=command] [--help] [--version] < name-list [>
archive]
```

```
cpio {-i|--extract} [-bcdmnrtsuvBSV] [-C bytes] [-E file] [-H format] [-M message] [-R
[user][:][group]] [-I [[user@]host:]archive] [-F [[user@]host:]archive] [--file=[[user@]host:]archive]
[--make-directories] [--nonmatching] [--preserve-modification-time] [--numeric-uid-gid] [--rename]
[-t|--list] [--swap-bytes] [--swap] [--dot] [--unconditional] [--verbose] [--block-size=blocks]
[--swap-halfwords] [--io-size=bytes] [--pattern-file=file] [--format=format]
[--owner=[user][:][group]] [--no-preserve-owner] [--message=message] [--force-local] [--no-abso-
lute-filenames] [--sparse] [--only-verify-crc] [--quiet] [--rsh-command=command] [--help] [--ver-
sion] [pattern...] [< archive]
```

```
cpio {-p|--pass-through} [-0adlmuvLV] [-R [user][:][group]] [--null] [--reset-access-time] [--make-
directories] [--link] [--quiet] [--preserve-modification-time] [--unconditional] [--verbose] [--dot]
[--dereference] [--owner=[user][:][group]] [--no-preserve-owner] [--sparse] [--help] [--version] des-
tination-directory < name-list
```

DESCRIPTION

This manual page documents the GNU version of **cpio**. **cpio** copies files into or out of a cpio or tar archive, which is a file that contains other files plus information about them, such as their file name, owner, time-stamps, and access permissions. The archive can be another file on the disk, a magnetic tape, or a pipe. **cpio** has three operating modes.

In copy-out mode, **cpio** copies files into an archive. It reads a list of filenames, one per line, on the standard input, and writes the archive onto the standard output. A typical way to generate the list of filenames is with the **find** command; you should give **find** the **-depth** option to minimize problems with permissions on directories that are unwritable or not searchable.

In copy-in mode, **cpio** copies files out of an archive or lists the archive contents. It reads the archive from the standard input. Any non-option command line arguments are shell globbing patterns; only files in the archive whose names match one or more of those patterns are copied from the archive. Unlike in the shell, an initial **'.'** in a filename does match a wildcard at the start of a pattern, and a **'/'** in a filename can match wildcards. If no patterns are given, all files are extracted.

In copy-pass mode, **cpio** copies files from one directory tree to another, combining the copy-out and copy-in steps without actually using an archive. It reads the list of files to copy from the standard input; the directory into which it will copy them is given as a non-option argument.

cpio supports the following archive formats: binary, old ASCII, new ASCII, crc, HPUX binary, HPUX old ASCII, old tar, and POSIX.1 tar. The binary format is obsolete because it encodes information about the files in a way that is not portable between different machine architectures. The old ASCII format is portable between different machine architectures, but should not be used on file systems with more than 65536 i-nodes. The new ASCII format is portable between different machine architectures and can be used on any size file system, but is not supported by all versions of **cpio**; currently, it is only supported by GNU and Unix System V R4. The crc format is like the new ASCII format, but also contains a checksum for each file which **cpio** calculates when creating an archive and verifies when the file is extracted from the archive. The HPUX formats are provided for compatibility with HPUX's **cpio** which stores device files differently.

The tar format is provided for compatability with the **tar** program. It can not be used to archive files with names longer than 100 characters, and can not be used to archive "special" (block or character devices) files. The POSIX.1 tar format can not be used to archive files with names longer than 255 characters (less unless they have a **'/'** in just the right place).

By default, **cpio** creates binary format archives, for compatibility with older **cpio** programs. When extracting from archives, **cpio** automatically recognizes which kind of archive it is reading and can read archives created on machines with a different byte-order.

Some of the options to **cpio** apply only to certain operating modes; see the SYNOPSIS section for a list of which options are allowed in which modes.

OPTIONS

-0, --null

In copy-out and copy-pass modes, read a list of filenames terminated by a null character instead of a newline, so that files whose names contain newlines can be archived. GNU **find** is one way to produce a list of null-terminated filenames.

-a, --reset-access-time

Reset the access times of files after reading them, so that it does not look like they have just been read.

-A, --append

Append to an existing archive. Only works in copy-out mode. The archive must be a disk file specified with the **-O** or **-F (--file)** option.

-b, --swap

In copy-in mode, swap both halfwords of words and bytes of halfwords in the data. Equivalent to **-sS**. Use this option to convert 32-bit integers between big-endian and little-endian machines.

-B

Set the I/O block size to 5120 bytes. Initially the block size is 512 bytes.

--block-size=BLOCK-SIZE

Set the I/O block size to BLOCK-SIZE * 512 bytes.

-c

Use the old portable (ASCII) archive format.

-C IO-SIZE, --io-size=IO-SIZE

Set the I/O block size to IO-SIZE bytes.

-d, --make-directories

Create leading directories where needed.

-E FILE, --pattern-file=FILE

In copy-in mode, read additional patterns specifying filenames to extract or list from FILE. The lines of FILE are treated as if they had been non-option arguments to **cpio**.

-f, --nonmatching

Only copy files that do not match any of the given patterns.

-F, --file=archive

Archive filename to use instead of standard input or output. To use a tape drive on another machine as the archive, use a filename that starts with 'HOSTNAME:'. The hostname can be preceded by a username and an '@' to access the remote tape drive as that user, if you have permission to do so (typically an entry in that user's '~/.rhosts' file).

--force-local

With **-F**, **-I**, or **-O**, take the archive file name to be a local file even if it contains a colon, which would ordinarily indicate a remote host name.

-H FORMAT, --format=FORMAT

Use archive format FORMAT. The valid formats are listed below; the same names are also recognized in all-caps. The default in copy-in mode is to automatically detect the archive format, and in copy-out mode is "bin".

bin The obsolete binary format.

odc The old (POSIX.1) portable format.

- `newc` The new (SVR4) portable format, which supports file systems having more than 65536 i-nodes.
- `crc` The new (SVR4) portable format with a checksum added.
- `tar` The old tar format.
- `ustar` The POSIX.1 tar format. Also recognizes GNU **tar** archives, which are similar but not identical.
- `hpbm` The obsolete binary format used by HP-UX's `cpio` (which stores device files differently).
- `hpdc` The portable format used by HP-UX's `cpio` (which stores device files differently).
- `-i, --extract`
Run in copy-in mode.
- `-I archive`
Archive filename to use instead of standard input. To use a tape drive on another machine as the archive, use a filename that starts with 'HOSTNAME:'. The hostname can be preceded by a username and an '@' to access the remote tape drive as that user, if you have permission to do so (typically an entry in that user's '~/.rhosts' file).
- `-k` Ignored; for compatibility with other versions of **cpio**.
- `-l, --link`
Link files instead of copying them, when possible.
- `-L, --dereference`
Dereference symbolic links (copy the files that they point to instead of copying the links).
- `-m, --preserve-modification-time`
Retain previous file modification times when creating files.
- `-M MESSAGE, --message=MESSAGE`
Print MESSAGE when the end of a volume of the backup media (such as a tape or a floppy disk) is reached, to prompt the user to insert a new volume. If MESSAGE contains the string "%d", it is replaced by the current volume number (starting at 1).
- `-n, --numeric-uid-gid`
In the verbose table of contents listing, show numeric UID and GID instead of translating them into names. Also extracts tar archives using the numeric UID and GID instead of the user/group names. (**cpio** archives are always extracted using the numeric UID and GID.)
- `--no-absolute-filenames`
In copy-in mode, create all files relative to the current directory, even if they have an absolute file name in the archive.
- `--no-preserve-owner`
In copy-in mode and copy-pass mode, do not change the ownership of the files; leave them owned by the user extracting them. This is the default for non-root users, so that users on System V don't inadvertently give away files.
- `-o, --create`
Run in copy-out mode.
- `-O archive`
Archive filename to use instead of standard output. To use a tape drive on another machine as the archive, use a filename that starts with 'HOSTNAME:'. The hostname can be preceded by a username and an '@' to access the remote tape drive as that user, if you have permission to do so (typically an entry in that user's '~/.rhosts' file).
- `--only-verify-crc`
When reading a CRC format archive in copy-in mode, only verify the CRC's of each file in the archive, don't actually extract the files.

- p, --pass-through*
Run in copy-pass mode.
- quiet*
Do not print the number of blocks copied.
- r, --rename*
Interactively rename files.
- R [user][:][group], --owner [user][:][group]*
In copy-out and copy-pass modes, set the ownership of all files created to the specified user and/or group. Either the user or the group, or both, must be present. If the group is omitted but the ":" or "." separator is given, use the given user's login group. Only the super-user can change files' ownership.
- rsh-command=COMMAND*
Notifies **mt** that it should use **COMMAND** to communicate with remote devices instead of **/usr/bin/ssh** or **/usr/bin/rsh**.
- sparse*
In copy-in and copy-pass modes, write files with large blocks of zeros as sparse files.
- s, --swap-bytes*
In copy-in mode, swap the bytes of each halfword (pair of bytes) in the files.
- S, --swap-halfwords*
In copy-in mode, swap the halfwords of each word (4 bytes) in the files.
- t, --list*
Print a table of contents of the input.
- u, --unconditional*
Replace all files, without asking whether to replace existing newer files with older files.
- v, --verbose*
List the files processed, or with *-t*, give an 'ls -l' style table of contents listing. In a verbose table of contents of a ustar archive, user and group names in the archive that do not exist on the local system are replaced by the names that correspond locally to the numeric UID and GID stored in the archive.
- V --dot*
Print a "." for each file processed.
- version*
Print the **cpio** program version number and exit.

NAME

crfile – create a dump file with a certain size

SYNOPSIS

edrc/bin/crfile [**-h**]

crfile { **-k** | **-m** | **-g** } *size* **-f** *filename*

AVAILABILITY

WA2L/edrc

DESCRIPTION

Create a dump file with a size to be specified.

Ahead of writing the file, it is checked if the file size does not exceed the available space on the file system.

OPTIONS

-h usage message.

-k file *size* specified in kilobytes.

-m file *size* specified in megabytes.

-g file *size* specified in gigabytes.

size file size.

-f *filename*
 name of the dump file to be created.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 **crfile** cannot write to the file specified in the **-f** option.

- 2** operating system not supported, yet. See **osid(3)** if you get this error.
- 3** file size of dump file exceeds available filesystem space.
- 4** usage listed.
- 5** file creation aborted.
- 6** the **size** specified was not greater than 0.
- 7** file specified in the **-f** option already exists.
- 11** a temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

-

EXAMPLES

-

SEE ALSO

bdf(1), **df(1)**, **edrcintro(1)**,

NOTES

-

BUGS

-

AUTHOR

crfile was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

crond – daemon to execute scheduled commands (cron in WA2L/edrc)

SYNOPSIS

edrc/lib/crond [**-h** | **-i** | **-n** | **-p** | **-P** | **-s** | **-m** *mail-command*]

crond -x [ext,sch,proc,pars,load,misc,test,bit]

crond -V

DESCRIPTION

Crond is started from **edrcinit**(1m).

Crond searches **edrc/var/spool/cron/** for crontab files which are named after accounts in **/etc/passwd**. The found crontabs are loaded into the memory. **Crond** also searches for any files in the **edrc/var/spool/cron.d/** directory, which have a different format (see **crontab**(4)). **Crond** examines all stored crontabs and checks each job to see if it needs to be run in the current minute. When executing commands, any output is mailed to the owner of the crontab (or to the user specified in the **MAILTO** environment variable in the crontab, if such exists). Any job output can also be sent to syslog by using the **-s** option.

Crond checks its crontables' modtimes every minute to check for any changes and reloads the crontables which have changed. There is no need to restart **Crond** after some of the crontables were modified.

Crond checks these files and directories:

edrc/etc/crontab

system crontab. The file is empty/not used by default.

edrc/var/spool/cron.d/

directory that contains system cronjobs stored for different users.

edrc/var/spool/cron/

directory that contains user crontables created by the **ecrontab** command (this is the most commonly used job scheduling method).

Note that the **ecrontab**(1) command updates the modtime of the spool directory whenever it changes a crontab.

Daylight Saving Time and other time changes

Local time changes of less than three hours, such as those caused by the Daylight Saving Time changes, are handled in a special way. This only applies to jobs that run at a specific time and jobs that run with a granularity greater than one hour. Jobs that run more frequently are scheduled normally.

If time was adjusted one hour forward, those jobs that would have run in the interval that has been skipped will be run immediately. Conversely, if time was adjusted backward, running the same job twice is avoided.

Time changes of more than 3 hours are considered to be corrections to the clock or the timezone, and the new time is used immediately.

It is possible to use different time zones for crontables. See **crontab**(4) for more information.

OPTIONS

- h** Prints a help message and exits.
- i** Disables inotify support (inotify is not compiled in).
- m** This option allows you to specify a shell command to use for sending **Crond** mail output instead of using **sendmail(8)**. This command must accept a fully formatted mail message (with headers) on standard input and send it as a mail message to the recipients specified in the mail headers. Specifying the string **off** (i.e., **crond -m off**) will disable the sending of mail.
- n** Tells the daemon to run in the foreground. This can be useful when starting it out of init.
- f** the same as **-n**, consistent with other crond implementations.
- p** Allows **Crond** to accept any user set crontables.
- P** Don't set PATH. PATH is instead inherited from the environment.
- s** This option will direct **Crond** to send the job output to the system log using **syslog(3)**. This is useful if your system does not have **sendmail(8)**, installed or if mail is disabled.
- x** This option allows you to set debug flags.
- V** Print version and exit.

SIGNALS

When the *SIGHUP* is received, the **Crond** daemon will close and reopen its log file. This proves to be useful in scripts which rotate and age log files. Naturally, this is not relevant if **Crond** was built to use **syslog(3)**.

CAVEATS

All crontab files have to be regular files or symlinks to regular files, they must not be executable or writable for anyone else but the owner. This requirement can be overridden by using the **-p** option on the crond command line. If inotify support is in use, changes in the symlinked crontabs are not automatically noticed by the cron daemon. The cron daemon must receive a SIGHUP signal to reload the crontabs. This is a limitation of the inotify API.

The syslog output will be used instead of mail, when sendmail is not installed.

SEE ALSO

edrcintro(1), **crontab(4)**, **ecrontab(1)**, **ecronnext(1)**, **edrcinit(1m)**, **sendmail(8)**, **syslog(3)**

AUTHOR

Paul Vixie <vixie@isc.org>
Marcela Maslanova <mmlasano@redhat.com>
Colin Dean <colin@colin-dean.org>
Tomas Mraz <tmraz@fedoraproject.org>

NAME

cronhandler – handle cron and at in a cluster environment

SYNOPSIS

edrc/bin/cronhandler [**-h**]

cronhandler -a { save | load | start } -s *save_dir* -u { *userlist* | *homedir* }

cronhandler -a stop -s *save_dir*

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **cronhandler** it is possible to transfer **crontab** and **at** entries from one cluster node to another. This is important if users related to a MC/ServiceGuard package are allowed to use **cron** and **at**.

cronhandler takes care of the user's **crontab** entries and transfers even defined **at** jobs to the other cluster node. Furthermore all **cron** and **at** control files (**cron.allow**, **cron.deny**, **at.allow**, **at.deny**) are maintained properly. Therefore it is ensured that a user related to a package can only define **crontab** and **at** entries if this user logs on to the node where the package is up and running (i.e. where the **crontab** entries and **at** jobs are loaded).

The basic concept of **cronhandler** is to save all **crontab** and **at** entries to a set of files located on a filesystem related to a MC/ServiceGuard package.

After the package switch the saved files are loaded and applied to the system where the package is starting. **crontab** entries are loaded like they were defined on the source node. If the execution of an **at** job falls into the short period of time between the package stop and a restart, the job will be rescheduled (current time + 60 seconds contingency time) on the target node. If there is a job planned on the target node at the same time a job from the source node likes to run, the job from the source node will be rescheduled one second after an already existing job.

Failover mechanism:

Preparation

- + permanently write cron/at entries to the *save_dir*
(**cronhandler** started as daemon)

Failover situation

- (1) save cron/at entries to the *save_dir*
(**cronhandler -a save ...**).
- (2) stop cluster package on current node
- 3 start cluster package on alternate node
- 4 load cron/at entries from *save_dir*
(**cronhandler -a load ...**)

() = step will only happen if the cluster node can be shut down properly.

After a **save** or **load** the **cron** daemon is restarted to ensure that all possibly changed **crontab** definitions and **at** jobs are recognized again.

OPTIONS

-h usage message.

-a action:

save

save **crontab** entries and current **at** jobs of user(s) specified with the **-u** option to the directory specified with the **-s** option. After a save of the **crontab** entries and all current **at** jobs, the **crontab** and **at** jobs are removed from the system and the related user is no longer allowed to define **crontab** and **at** entries on the node. **save** is normally used to save **crontab** and **at** entries during a controlled cluster switch while halting the cluster package. After invoking **-a save** the **cronhandler** daemon is stopped.

load

load a previous saved **crontab** and **at** specification and allow the related user **crontab** and **at** definition where granted. After loading the saved jobs **cronhandler** enters into daemon mode and continues to run in the background. In daemon mode **cronhandler** writes all **crontab** entries and current **at** jobs of user(s) specified with the **-u** option in a regular interval to the directory specified with the **-s** option. The write interval can be specified in the config file **edrc/etc/cronhandler.cfg** with the setting **WRITE_INTERVAL**.

stop

stop the **cronhandler** daemon. Normally you don't have to stop the daemon by hand, it is stopped automatically after the **save** action has been invoked. The **stop** option is provided for ease of administration during maintenance tasks on the system when you don't want to have the **cronhandler** daemon running, but all **cron** and **at** jobs must continue to run.

start

start the **cronhandler** daemon. Normally you don't have to start the daemon by hand, it is started automatically after the **load** action has been invoked. The **start** option is provided for ease of administration during maintenance tasks on the system when you don't want to have the **cronhandler** daemon running, but all **cron** and **at** jobs must continue to run.

write

this action does no longer exist as command line option and is ignored if issued. It has no side effects (except for a logfile entry) if this option is still used.

-s *save_dir*

directory where to save all **crontab** and **at** data. See section **FILES** for the recommended permissions of this directory.

-u

userlist a list of users separated by commas. The **crontab** and **at** entries of those users will be saved to the directory specified by the **-s** option.

homedir

if you have many users or if you like to react **cronhandler** dynamically on new/removed users you should use the definition of a \$HOME directory. All subdirectories of the *homedir* will be considered as users. Therefore the username has to match with the \$HOME (what is normally the case). See section **EXAMPLES** for a common setup.

SIGNALS

The following signals are handled by **cronhandler**. Do not use other signals as those listed below, as long as you do not really know what you are doing and what the consequences are. In general there is no need to invoke those signals by your own, this signals are used for inter process communication of **cronhandler**.

- INT** kill the process with **kill -SIGINT PID** . Issuing this signal, the daemon will end.
- TERM** kill the process without an argument (**kill PID**). Issuing this signal, the daemon will end.

EXIT STATUS

- 0** no error.
- 2** operating system is not supported. See **osid(3)** if you get this error.
- 4** usage listed.
- 7** the **cronhandler** has been started with the depreciated command line option **-a write**. This option is ignored from command line.
- 11** a temporary directory could not be claimed or created in **/var/tmp**. Check the system temporary directory **/var/tmp** if you get this error, it is an indicator of system intrusion.

FILES**etc/cronhandler.cfg**

configuration file of **cronhandler**, see **cronhandler(4)** for more information. The configfile is reloaded in the interval defined in **WRITE_INTERVAL**, therefore all settings (except **LOCKDIR**) can be changed without stopping the **cronhandler** daemon.

var/log/cronhandler.log

logfile of **cronhandler**.

<save_dir>/

directory where to save the **cron** and **at** definitions. Due to security reasons the directory permissions should be root:sys, 0751 . You should not use this directory to save other data because it is maintained completely by **cronhandler**.

<save_dir>/cronhandler/action=save

flagfile if the last **cronhandler** action was **save**. This file contains the nodename where the action has been executed.

<save_dir>/cronhandler/action=write

flagfile if the last **cronhandler** action was **write**. This file contains the nodename where the action has been executed.

<save_dir>/cronhandler/<username>/cron/<username>

saved crontab.

<save_dir>/cronhandler/<username>/cron.allow=YES

flagfile if user was defined in **cron.allow** on source node.

<save_dir>/cronhandler/<username>/cron.allow=NO

flagfile if user was not defined in **cron.allow** on source node.

<save_dir>/cronhandler/<username>/at.allow=YES

flagfile if user was defined in **at.allow** on source node.

<save_dir>/cronhandler/<username>/at.allow=NO

flagfile if user was not defined in **at.allow** on source node.

<save_dir>/cronhandler/<username>/at/

directory to save currently defined **at** jobs for the user.

<homedir>/

user's \$HOME directory.

/usr/lib/cron/cron.allow /usr/lib/cron/cron.deny

List of users which are allowed/denied to create a **crontab**(1) on HP-UX and Solaris.

/etc/cron.allow /etc/cron.deny

List of users which are allowed/denied to create a **crontab**(1) on Linux.

/sbin/init.d/cron

Start/Stop script for **cron**(1m) on HP-UX.

/etc/init.d/crond

Start/Stop script for **cron**(8) on Linux and on Solaris.

/var/spool/cron/atjobs

at job spool dir accessed on Solaris only.

EXAMPLES

1) Common setup using MC/ServiceGuard

In this example the users `sys_asys` and `ora_asys` are related to a MC/ServiceGuard (see also <http://www.hp.com>) package. Both users are allowed to create at jobs, but only the user `ora_asys` is allowed to have an own crontab. The filesystem `/data_asys1` goes with the package (=moving disks) and is therefore used to carry the cron entries and at jobs from one to the other node.

1.1) `/etc/passwd` (on all nodes):

```
:
:
ora_asys:adSKflweIRsdf:253:101::/data_asys1/home/ora_asys:/bin/ksh
sys_asys:Wm9MyTyKtRI2c:106:104::/data_asys1/home/sys_asys:/sbin/sh
:
:
```

1.2) `/usr/lib/cron/cron.allow` (on active node):

```
:
ora_asys
:
```

1.3) `/usr/lib/cron/at.allow` (on active node):

```
:
ora_asys
sys_asys
:
```

1.4) `cd /data_asys1/home; ls -ald` (on active node):

```
drwxr-xr-x  6 ora_asys oinstall ... ora_asys/
drwxrwxr-x 10 sys_asys autosys  ... sys_asys/
```

HINT: `/data_asys1` goes with the asys package.

1.5) `/etc/cmcluster/asys_sv1_prod/asys_sv1_prod.cntl` (on all nodes):

```
:
:

function customer_defined_run_cmds
{
    # ADD customer defined run commands.

    :
    :

    /opt/edrc/bin/cronhandler -a load \
        -s /data_asys1/pkg_cron -u /data_asys1/home

    test_return 51
}
```

```

:
:

function customer_defined_halt_cmds
{
    # ADD customer defined halt commands.

    /opt/edrc/bin/cronhandler -a save \
        -s /data_asys1/pkg_cron -u /data_asys1/home

    :
    :

    test_return 52
}

:
:

```

1.6) Initial steps to initiate the daemon to work (on active node):

Save the current **crontab** definitions and **at** jobs to disk:

/opt/edrc/bin/cronhandler -a save -s /data_asys1/pkg_cron -u /data_asys1/home

```

cronhandler - handle cron and at entries in a cluster environment,

stop cronhandler daemon for '/data_asys1/pkg_cron' ...(not running)
save crontab and at entries ...
user: asys_sv1 ...
    clear old cron/at saves ... done.
    disable cron execution ... done.
    save at jobs ...
    done.
    disable at execution ... done.
done.
user: ora_sys ...
    clear old cron/at saves ... done.
    save cron entries ... done.
    remove cron entries ... done.
    disable cron execution ... done.
    save at jobs ...
        save at job 1189778400.a ... done.
        remove at job 1189778400.a ... done.
    done.
    disable at execution ... done.
done.
user: sys_asys ...
    clear old cron/at saves ... done.
    disable cron execution ... done.
    save at jobs ...
        save at job 1189778500.a ... done.
        remove at job 1189778500.a ... done.

```

```

        save at job 1189779100.a ... done.
        remove at job 1189779100.a ... done.
    done.
    disable at execution ... done.
done.
clean up remaining saves (corpses) ...
done.
restart cron ...(wait 5 seconds)... done.
done.

```

Immediately load the saved **crontab** definitions and **at** jobs from disk back into the system:

/opt/edrc/bin/cronhandler -a load -s /data_asys1/pkg_cron -u /data_asys1/home

```

cronhandler - handle cron and at entries in a cluster environment,

load crontab and at entries ...
    asys_sv1 ...
        cron usage not authorized
        at usage not authorized
    done.
    ora_sys ...
        enable cron execution ... done.
        load cron entries ... done.
        enable at execution ... done.
        load at jobs ...
            load at job 1189778400.a ... done.
        done.
    done.
    sys_asys ...
        cron usage not authorized
        enable at execution ... done.
        load at jobs ...
            load at job 1189778500.a ... done.
            load at job 1189779100.a ... done.
        done.
    done.
    restart cron ...(wait 5 seconds)... done.
done.
start cronhandler daemon for '/data_asys1/pkg_cron' ...(PID=18790).

```

From now on the **cronhandler** daemon writes **crontab** definitions and defined **at** jobs in a regular interval to disk and no manual interventions are needed. The correct stop and start is handled via the MC/ServiceGuard control script (**/etc/cmcluster/asys_sv1_prod/asys_sv1_prod.cntl**).

SEE ALSO

edrcintro(1), **osid(3)**, **cron(1m)**, **at(1)**, **crontab(1)**, **cronhandler.cfg(4)**,
Managing MC/ServiceGuard, March 2002, Hewlett-Packard Company, Part No: B3936-90055"

NOTES

Currently **cronhandler** runs on HP-UX (tested on 11.00 and 11.11), Linux (tested on RedHat 7.2) Solaris (tested on SunOS 5.6). Enhancements to support other operating systems might be realized on request (please ask the author).

In an earlier version of **cronhandler** a root **crontab** entry was needed to write the **crontab** definitions and the **at** jobs to disk - this is no longer the case. Eventually existing **/opt/edrc/bin/cronhandler -a write ...** calls are ignored and do not influence the **cronhandler**. To avoid logfile entries, the old **crontab** entries should be removed eventually.

Use **cron.allow** and **at.allow** to control **crontab** and **at** job definition.

BUGS

-

AUTHOR

cronhandler was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

cronhandler.cfg – configuration file for cronhandler

SYNOPSIS

edrc/etc/cronhandler.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **cronhandler** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **cronhandler**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

LOCKDIR

Lock dir of **cronhandler**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation. In General it is not recommended to set the lockdir within EDRC, locate it in a system own directory. This is the only setting that is ignored when the **cronhandler.cfg** file is reloaded.

Example: LOCKDIR=/var/run/cronhandler

Default: LOCKDIR=var/lock

WRITE_INTERVAL

Interval in seconds of writing the **crontab** definitions and **at** jobs to disk. This is also the interval in which the configfile is reloaded.

Example: WRITE_INTERVAL=120

Default: WRITE_INTERVAL=60

CRON_START_DELAY

Delay in seconds between **cron** stop and **cron** start after a **save** or **load** action.

Example: CRON_START_DELAY=10

Default: CRON_START_DELAY=5

CONTINGENCY

Contingency time in seconds when a **at** job has to be rescheduled. A reschedule occurs when a **at** job falls into the time between a package stop (**save** action) and restart (**load** action).

Example: CONTINGENCY=90

Default: CONTINGENCY=60

SEE ALSO

cronhandler(1m), **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

cronhandler.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

crontab – files used to schedule the execution of programs (cron in WA2L/edrc)

DESCRIPTION

A **crontab** file contains instructions for the **crond**(3) daemon in the following simplified manner: "run this command at this time on this date". Each user can define their own crontab. Commands defined in any given crontab are executed under the user who owns that particular crontab. Uucp and News usually have their own crontabs, eliminating the need for explicitly running **su**(1) as part of a cron command.

Blank lines, leading spaces, and tabs are ignored. Lines whose first non-white space character is a pound-sign (#) are comments, and are not processed. Note that comments are not allowed on the same line as cron commands, since they are considered a part of the command. Similarly, comments are not allowed on the same line as environment variable settings.

An active line in a crontab is either an environment setting or a cron command. An environment setting is of the form:

```
name = value
```

where the white spaces around the equal-sign (=) are optional, and any subsequent non-leading white spaces in *value* is a part of the value assigned to *name*. The *value* string may be placed in quotes (single or double, but matching) to preserve leading or trailing white spaces.

Several environment variables are set up automatically by the **crond**(3) daemon. *SHELL* is set to **/bin/sh**, and *LOGNAME* and *HOME* are set from the **/etc/passwd** line of the crontab's owner. *HOME* and *SHELL* can be overridden by settings in the crontab; *LOGNAME* can not.

(Note: the *LOGNAME* variable is sometimes called *USER* on BSD systems and is also automatically set).

In addition to *LOGNAME*, *HOME*, and *SHELL*, **crond**(3) looks at the *MAILTO* variable if a mail needs to be sent as a result of running any commands in that particular crontab. If *MAILTO* is defined (and non-empty), mail is sent to the specified address. If *MAILTO* is defined but empty (*MAILTO=""*), no mail is sent. Otherwise, mail is sent to the owner of the crontab. This option is useful if you decide to use **/bin/mail** instead of **/usr/lib/sendmail** as your mailer. Note that **/bin/mail** does not provide aliasing and UUCP usually does not read its mail. If *MAILFROM* is defined (and non-empty), it is used as the envelope sender address, otherwise, "root" is used.

(Note: Both *MAILFROM* and *MAILTO* variables are expanded, so setting them as in the following example works as expected: *MAILFROM=cron-\$USER@cron.com* (*\$USER* is replaced by the system user))

By default, cron sends a mail using the 'Content-Type:' header of 'text/plain' with the 'charset=' parameter set to the 'charmap/codeset' of the locale in which **crond**(3) is started up, i.e., either the default system locale, if no *LC_** environment variables are set, or the locale specified by the *LC_** environment variables (see *locale*(7)). Different character encodings can be used for mailing cron job outputs by setting the *CONTENT_TYPE* and *CONTENT_TRANSFER_ENCODING* variables in a crontab to the correct values of the mail headers of those names.

The *CRON_TZ* variable specifies the time zone specific for the cron table. The user should enter a time according to the specified time zone into the table. The time used for writing into a log file is taken from the local time zone, where the daemon is running.

The *RANDOM_DELAY* variable allows delaying job startups by random amount of minutes with upper limit specified by the variable. The random scaling factor is determined during the cron daemon startup so it remains constant for the whole run time of the daemon.

The format of a cron command is similar to the V7 standard, with a number of upward-compatible extensions. Each line has five time-and-date fields followed by a **username** (if this is the **system** crontab file), and followed by a command. Commands are executed by **crond**(3) when the 'minute', 'hour', and 'month of the year' fields match the current time, *and* at least one of the two 'day' fields ('day of month', or 'day of

week') match the current time (see "Note" below).

Note that this means that non-existent times, such as the "missing hours" during the daylight savings time conversion, will never match, causing jobs scheduled during the "missing times" not to be run. Similarly, times that occur more than once (again, during the daylight savings time conversion) will cause matching jobs to be run twice.

crond(3) examines cron entries every minute.

The time and date fields are:

| field | allowed values |
|--------------|--------------------------------------|
| ----- | ----- |
| minute | 0-59 |
| hour | 0-23 |
| day of month | 1-31 |
| month | 1-12 (or names, see below) |
| day of week | 0-7 (0 or 7 is Sunday, or use names) |

A field may contain an asterisk (*), which always stands for "first-last".

Ranges of numbers are allowed. Ranges are two numbers separated with a hyphen. The specified range is inclusive. For example, 8-11 for an 'hours' entry specifies execution at hours 8, 9, 10, and 11. The first number must be less than or equal to the second one.

Randomization of the execution time within a range can be used. A random number within a range specified as two numbers separated with a tilde is picked. The specified range is inclusive. For example, 6~15 for a 'minutes' entry picks a random minute within 6 to 15 range. The random number is picked when crontab file is parsed. The first number must be less than or equal to the second one. You might omit one or both of the numbers specifying the range. For example, ~ for a 'minutes' entry picks a random minute within 0 to 59 range.

Lists are allowed. A list is a set of numbers (or ranges) separated by commas. Examples: "1,2,5,9", "0-4,8-12".

Step values can be used in conjunction with ranges. Following a range with "/<number>" specifies skips of the number's value through the range. For example, "0-23/2" can be used in the 'hours' field to specify command execution for every other hour (the alternative in the V7 standard is "0,2,4,6,8,10,12,14,16,18,20,22"). Step values are also permitted after an asterisk, so if specifying a job to be run every two hours, you can use "*/2".

Names can also be used for the 'month' and 'day of week' fields. Use the first three letters of the particular day or month (case does not matter). Ranges and lists of names are allowed. Examples: "mon,wed,fri", "jan-mar".

If the UID of the owner is 0 (root), the first character of a crontab entry can be "-" character. This will prevent cron from writing a syslog message about the command being executed.

The "sixth" field (the rest of the line) specifies the command to be run. The entire command portion of the line, up to a newline or a "%" character, will be executed by **/bin/sh** or by the shell specified in the SHELL variable of the cronfile. A "%" character in the command, unless escaped with a backslash (\), will be changed into newline characters, and all data after the first % will be sent to the command as standard input.

Note: The day of a command's execution can be specified in the following two fields — 'day of month', and 'day of week'. If both fields are restricted (i.e., do not contain the "*" character), the command will be run when *either* field matches the current time. For example,

"30 4 1,15 * 5" would cause a command to be run at 4:30 am on the 1st and 15th of each month, plus every Friday.

A crontab file syntax can be tested before an install using the -T option. See **ecrontab(1)** for details.

EXAMPLE CRON FILE

```
# use /bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to 'paul', no matter whose crontab this is
MAILTO=paul
#
CRON_TZ=Japan
# run five minutes after midnight, every day
5 0 * * * $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 2:15pm on the first of every month -- output mailed to paul
15 14 1 * * $HOME/bin/monthly
# run at 10 pm on weekdays, annoy Joe
0 22 * * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%
23 0-23/2 * * * echo "run 23 minutes after midn, 2am, 4am ..., everyday"
5 4 * * sun echo "run at 5 after 4 every sunday"
```

FILES

edrc/etc/crontab
main system crontab file.

edrc/var/spool/cron/
a directory for storing crontabs defined by users.

edrc/var/spool/cron.d/
a directory for storing system crontabs.

SEE ALSO

edrcintro(1), **crond(3)**, **ecronnext(1)**, **ecrontab(1)**

EXTENSIONS

These special time specification "nicknames" which replace the 5 initial time and date fields, and are prefixed with the '@' character, are supported:

```
@reboot : Run once after reboot.
@yearly : Run once a year, ie. "0 0 1 1 *".
```

@annually : Run once a year, ie. "0 0 1 1 *".
@monthly : Run once a month, ie. "0 0 1 * *".
@weekly : Run once a week, ie. "0 0 * * 0".
@daily : Run once a day, ie. "0 0 * * *".
@hourly : Run once an hour, ie. "0 * * * *".

CAVEATS

crontab files have to be regular files or symlinks to regular files, they must not be executable or writable for anyone else but the owner. This requirement can be overridden by using the **-p** option on the **crond** command line. If inotify support is in use, changes in the symlinked crontabs are not automatically noticed by the cron daemon. The cron daemon must receive a SIGHUP signal to reload the crontabs. This is a limitation of the inotify API.

cron requires that each entry in a crontab end in a newline character. If the last entry in a crontab is missing a newline (i.e. terminated by EOF), cron will consider the crontab (at least partially) broken. A warning will be written to syslog.

AUTHOR

Paul Vixie <vixie@isc.org>

NAME

crypt – encode/decode files

SYNOPSIS

[*password*]

DESCRIPTION

reads from the standard input and writes on the standard output. *password* is a key that selects a particular transformation. If no *password* is given, demands a key from the terminal and turns off printing while the key is being typed in. encrypts and decrypts with the same key:

The latter command decrypts the file and prints the clear version.

Files encrypted by are compatible with those treated by the editor in encryption mode (see *ed*(1)).

Security of encrypted files depends on three factors: the fundamental method must be hard to solve; direct search of the key space must be infeasible; “sneak paths” by which keys or clear text can become visible must be minimized.

implements a one-rotor machine designed along the lines of the German Enigma, but with a 256-element rotor. Methods of attack on such machines are widely known; thus crypt provides minimal security.

The transformation of a key into the internal settings of the machine is deliberately designed to be expensive; i.e., to take a substantial fraction of a second to compute. However, if keys are restricted to, for example, three lowercase letters, then encrypted files can be read by expending only a substantial fraction of five minutes of machine time.

Since the key is an argument to the command, it is potentially visible to users executing the or a derivative (see *ps*(1)). The choice of keys and key security are the most vulnerable aspect of

EXAMPLES

The following example demonstrates the use of to edit a file that the user wants to keep strictly confidential:

```
...
...
...
```

Note that the option is the encryption mode of and prompts the user for the same key with which the file was encrypted.

WARNINGS

If output is piped to and the encryption key is given on the command line, can leave terminal modes in a strange state (see *nroff*(1) and *stty*(1)).

If two or more files encrypted with the same key are concatenated and an attempt is made to decrypt the result, only the the first of the original files is decrypted correctly.

FILES

for typed key

SEE ALSO

ed(1), *makekey*(1), *stty*(1).

NAME

csv – filter to process CSV data

SYNOPSIS

edrc/lib/csv [**-h**]

cat *csv_data* | **csv** *command* [*options*]

csv *command* [*options*] < *csv_data*

csv c2s

csv t2s

csv w2s

csv addcol *COLNAME* ["*column data*"]

csv cat

csv cat:authorized_keys

csv cat:crontab

csv cat:fstab

csv cat:group

csv cat:known_hosts

csv cat:mtab

csv cat:oratab

csv cat:passwd

csv cat:protocols

csv cat:rpc

csv cat:services

csv cat:shadow

csv cat:syscrontab

csv cat:mailcap

csv cat:mailcap:nokey

csv cat:datalist.dat

csv cat:edrcinit.cfg

csv cat:filedist.block

csv cat:fssum.cfg

csv cat:hostlist.dat

csv cat:lgcheckd.cfg

csv cat:logcheckd.cfg

csv cat:name_index

csv cat:osid.dat

csv cat:osid.dat.WA2L

csv cat:pscount.kp

csv cat:role_option.cfg

csv cat:schedule.dat

csv cat:timezone.dat

csv cat:volume.dat

csv cat:list
csv cat:name
csv cat:revision
csv cat:usage

csv cat:avatar
csv cat:mccli

csv comm *file*
csv comm:both *file*
csv comm:stdin *file*
csv comm:file *file*

csv concat *file...*

csv count

csv data

csv diff *file*

csv header

csv head *lines*

csv join *file* [*joinfieldnum1* [*joinfieldnum2*]]

csv list
csv list:nocount
csv list:noemptycols
csv list:noemptycols:nocount
csv list:noemptyrows
csv list:noemptyrows:nocount
csv list:noempty
csv list:noempty:nocount

csv lscol

csv md5 [*COLNAME*]

csv number [*COLNAME*]

csv order "*COLNAME1;COLNAME2; ... ;COLNAME_n*"

csv outer *file* [*joinfieldnum1* [*joinfieldnum2* [*unpairablesfilenum*]]]

csv record
csv record:nocount

csv renamecol "*OLDCOL1;OLDCOL2; ... ;OLDCOL_n*" "*NEWCOL1;NEWCOL; ... ;NEWCOL_n*"

csv row *linenumber*

csv sed "*sed script*"

csv sed:silent "*sed script*"

csv select "*COLNAME1;COLNAME2; ... ;COLNAME*n**"

csv sort [*sort_options*]

csv tac

csv tail *lines*

csv timestamp [*COLNAME*]

csv transpose

csv unselect "*COLNAME1;COLNAME2; ... ;COLNAME*n**"

csv variable [*prefix*]

csv variable:array [*prefix*]

csv variable:number [*prefix*]

csv where "*regex*"

csv where:ignorecase "*regex*"

csv wherenot "*regex*"

csv wherenot:ignorecase "*regex*"

csv match "*wildcard*"

csv match:ignorecase "*wildcard*"

csv matchnot "*wildcard*"

csv matchnot:ignorecase "*wildcard*"

AVAILABILITY

WA2L/edrc

DESCRIPTION

filter to process CSV data.

The field separator of the CSV data is a semicolon (;).

If comma (,) separated data needs to be processed, first use **csv c2s**, if tab separated data needs to be processed, first use **csv t2s** and if whitespace separated data needs to be processed, first use the **csv w2s** command to convert the comma separated data to semicolon separated.

OPTIONS

-h usage.

csv command [*options*]
 command with options to use with the CSV data stream. Description of available commands, see below.

csv addcol *COLNAME* ["*column data*"]
 add a column with the name *COLNAME* and the given "*column data*".

csv c2s convert comma separations to semicolon separations.

csv cat clean and join broken CSV lines.

csv cat:authorized_keys
 cat data with '~user/.ssh/authorized_keys' format as CSV.

csv cat:fstab
 cat data with '/etc/fstab' format as CSV.

csv cat:group
 cat data with '/etc/group' format as CSV.

csv cat:known_hosts
 cat data with '~user/.ssh/known_hosts' format as CSV.

csv cat:mtab
 cat data with '/etc/mtab' format as CSV.

csv cat:crontab
 cat data with user '/var/spool/cron/(...)/crontab' format as CSV.

csv cat:syscrontab
 cat data with system wide '/etc/crontab' format as CSV.

csv cat:mailcap
 cat data with '/etc/mailcap' format as CSV.

csv cat:mailcap:nokey
 cat data with '/etc/mailcap' format as CSV without keys in field data.

csv cat:oratab
 cat data with 'oratab' format as CSV.

csv cat:passwd
 cat data with '/etc/passwd' format as CSV.

csv cat:protocols

cat data with **'/etc/protocols'** format as CSV.

csv cat:rpc

cat data with **'/etc/rpc'** format as CSV.

csv cat:services

cat data with **'/etc/services'** format as CSV.

csv cat:shadow

cat data with **'/etc/shadow'** format as CSV.

csv cat:datalist.dat

cat data with **'edrc/var/lots/object/datalist.dat'** format as CSV.

csv cat:edrcinit.cfg

cat data with **'edrc/etc/edrcinit.cfg'** format as CSV.

csv cat:filedist.block

cat data with **'edrc/etc/filedist.block'** format as CSV.

csv cat:fssum.cfg

cat data with **'edrc/etc/fssum.cfg'** format as CSV.

csv cat:hostlist.dat

cat data with **'edrc/etc/hostlist.dat'** format as CSV.

csv cat:lgcheckd.cfg

cat data with **'edrc/etc/lgcheckd.cfg'** format as CSV.

csv cat:logcheckd.cfg

cat data with **'edrc/etc/logcheckd.cfg'** format as CSV.

csv cat:name_index

cat data with **'..name_index'** format as CSV.

csv cat:osid.dat

cat data with **'edrc/etc/osid.dat'** format as CSV.

csv cat:osid.dat.WA2L

cat data with **'edrc/etc/osid.dat.WA2L'** format as CSV.

csv cat:pscount.kp

cat data with **'edrc/var/pscount/*/ <app>.kp'** format as CSV.

csv cat:role_option.cfg

cat data with '**edrc/etc/role_option.cfg**' format as CSV.

csv cat:schedule.dat

cat data with '**edrc/var/lots/object/schedule.dat**' format as CSV.

csv cat:timezone.dat

cat data with '**edrc/etc/timezone.dat**' format as CSV.

csv cat:volume.dat

cat data with '**edrc/var/lots/object/volume.dat**' format as CSV.

csv cat:list cat data with **csv list** command output format as CSV.

csv cat:name

cat data with **name(1)** command output format as CSV.

csv cat:revision

cat data with **revision(1)** command output format as CSV.

csv cat:usage

cat data with **usage(1)** command output format as CSV.

csv cat:avtar

cat data with Avamar **avtar** command output format as CSV.

csv cat:mccli

cat data with Avamar **mccli** command output format as CSV.

csv comm file

compare **stdin** with a *file*, rows common to both (**stdin** and *file*).

The data from **stdin** and the *file*, is sorted by the **csv comm** command prior to comparing it.

csv comm:both file

compare **stdin** with a *file*, rows common to both (**stdin** and *file*).

The data from **stdin** and the *file*, is sorted by the **csv comm:both** command prior to comparing it.

csv comm:stdin file

compare **stdin** with a *file*, rows unique to **stdin**.

The data from **stdin** and the *file*, is sorted by the **csv comm:stdin** command prior to comparing it.

csv comm:file file

compare **stdin** with a *file*, rows unique to *file*.

The data from **stdin** and the *file*, is sorted by the **csv comm:file** command prior to comparing it.

csv concat *file...*

concatenate **stdin** with a space separated list of *files*.

csv count count number of data rows.

csv diff *file*

compare **stdin** with a *file*.

The data from **stdin** and the *file*, is sorted by the **csv diff** command prior to comparing it.

The file indicators <, > and = as known from the **diff**(1) command on normal files, are shown in an additional column with the name **DIFF**.

csv data print CSV data.

csv head *lines*

print *lines* lines from the beginning of CSV data.

csv header

print CSV header.

csv join *file* [*joinfieldnum1* [*joinfieldnum2*]]

join **stdin** with a file.

csv list print data as formatted list.

csv list:nocount

print data as formatted list without output of the count line.

csv list:noemptycols

print data as formatted list but do not list columns that have no data (which are empty).

csv list:noemptycols:nocount

print data as formatted list but do not list columns that have no data (which are empty) without output of the count line.

csv list:noemptyrows

print data as formatted list but do not list rows that have no data (which are empty).

csv list:noemptyrows:nocount

print data as formatted list but do not list rows that have no data (which are empty) without output of the count line.

csv list:noempty

print data as formatted list but do not list rows and columns that have no data (which are empty).

csv list:noempty:nocount

print data as formatted list but do not list rows and columns that have no data (which are empty) without output of the count line.

csv lscol list columns of CSV header as numbered list.

csv md5 [*COLNAME*]

add a md5 checksum column and optionally name the **MD5** column with *COLNAME*.

csv number [*COLNAME*]

number rows and optionally name the **NR** column with *COLNAME*.

csv order "*COLNAME1;COLNAME2; ... ;COLNAME_n*"

change order of columns.

csv outer *file* [*joinfieldnum1* [*joinfieldnum2* [*unpairablesfilenum*]]]

join **stdin** with a file including unpairable lines.

csv record print CSV line as record.

csv record:nocount

print CSV line as record without output of the count line.

csv renamecol "*OLDCOL1;OLDCOL2; ... ;OLDCOL_n*" "*NEWCOL1;NEWCOL2; ... ;NEWCOL_n*"

rename columns.

csv row *linenumber*

print data of record with *linenumber* from the CSV data.

csv sed "*sed script*"

stream editor (**sed**) for csv data.

csv sed:silent "*sed script*"

stream editor (**sed**) for csv data without automatic printing of pattern space (equals to **sed -n** "*sed script*").

csv select "*COLNAME1;COLNAME2; ... ;COLNAME_n*"

select columns.

csv sort [*sort_options*]

sort data.

csv tac print CSV data in reverse.

csv tail *lines*

print *lines* lines from the end of CSV data.

- csv timestamp** [*COLNAME*]
add a timestamp column in the military format *YYYY-MM-DD HH:MM:SS* and optionally name the **TIMESTAMP** column with *COLNAME*.
- csv transpose**
transpose the CSV file.
- csv t2s** convert tabulator separations to semicolon separations.
- csv unselect** "*COLNAME1;COLNAME2; ... ;COLNAME_n*"
de-select columns.
- csv variable** [*prefix*]
print CSV line as initialize-able variables **COLNAME="field"** or *prefix***COLNAME="field"**.
- csv variable:array** [*prefix*]
print CSV line as initialize-able numbered variables **COLNAME[#]="field"** or *prefix***COLNAME[#]="field"**.
- csv variable:number** [*prefix*]
print CSV line as initialize-able variables array **COLNAME_#="field"** or *prefix***COLNAME_#="field"**.
- csv where** "*regex*"
select data with a regular expression.
- csv where:ignorecase** "*regex*"
select data non case sensitive with a regular expression.
- csv wherenot** "*regex*"
inverse data select with a regular expression.
- csv wherenot:ignorecase** "*regex*"
inverse data select non case sensitive with a regular expression.
- csv match** "*wildcard*"
select data with a wildcard expression.
- csv match:ignorecase** "*wildcard*"
select data non case sensitive with a wildcard expression.
- csv matchnot** "*wildcard*"
inverse data select with a wildcard expression.
- csv matchnot:ignorecase** "*wildcard*"
inverse data select non case sensitive with a wildcard expression.

csv w2s convert whitespace separations to semicolon separations.

ENVIRONMENT

\$TMPDIR

temporary base directory (default=**/tmp/**) needed to sort data.

If file space shortages are observed set this variable to a location with sufficient space.

EXIT STATUS

0 no error.

1 error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), authorized_keys(8), comm(1), csvcat(3), csvq(3), csv2worksheet(3), crontab(1), crontab(5), diff(1), egrep(1), fit(3), fit2width(3), fnmatch(3), fstab(5), getmntent(3), group(5), head(1), indent(3), join(1), jq(3), mailcap(5), name(1), passwd(5), print_list(3), print_header(3), print_index(3), protocols(5), revision(1), sed(1), select_columns(3), services(5), shadow(5), sort(1), tac(1), tail(1), usage(1), xml2json(1)

NOTES

-

BUGS

-

AUTHOR

csv was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

csvcat – clean, join and cat csv file or stream

SYNOPSIS

edrc/lib/csvcat [**-h**]

csvcat *file...*

cat *file...* | **csvcat**

AVAILABILITY

WA2L/edrc

DESCRIPTION

prepare and print a CSV file or stream for further processing.

Comment lines that begin with a hash (#) and empty lines are eliminated.

If the first line is **sep=...** (as seen in some programatically produced **CSV** files) this line is removed.

Lines that are distributed over multiple lines thru a backslash (\) at the end of the line are joined.

White spaces before and after the field separator (;) are eliminated.

OPTIONS

-h usage message.

file... file(s) to be printed.

ENVIRONMENT

-

EXIT STATUS

0 no error.

x error codes of **sed**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **csv(3)**, **print_list(3)**, **sectioncat(3)**, **sed(3)**, **select_columns(3)**

NOTES

-

BUGS

-

AUTHOR

csvcat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

csvq – SQL-like query language for csv

SYNOPSIS

edrc/lib/csvq [**--help**]

csvq [*options*] [*subcommand*] ["*query*" | *argument*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

csvq is a command line tool to operate **CSV** files.

You can read, update, delete **CSV** records with SQL-like query.

You can also execute multiple operations sequentially in managed transactions by passing a procedure or using the interactive shell. In the multiple operations, you can use variables, cursors, temporary tables, and other features.

For the whole documentation see: <https://mithrandie.github.io/csvq/>

OPTIONS

--help print usage message.

csvq_options

See: <https://mithrandie.github.io/csvq/reference/command.html>

ENVIRONMENT

See: <https://mithrandie.github.io/csvq/reference/command.html#environment-configurations>

EXIT STATUS

See: https://mithrandie.github.io/csvq/reference/command.html#return_code

FILES

See: <https://mithrandie.github.io/csvq/>

EXAMPLES

See: <https://mithrandie.github.io/csvq/>

SEE ALSO

edrcintro(1) **csv(3)**, **jq(3)**, **xml2json(3)**, <https://mithrandie.github.io/csvq/>

NOTES

This manpage is a partial extract of the home page <https://mithrandie.github.io/csvq/> which has been written by Yuki Mithrandie <<https://github.com/mithrandie>>.

See the mentioned web page to view the complete **csvq** description.

BUGS

-

AUTHOR

csvq was developed by Yuki Mithrandie <<https://github.com/mithrandie>> (<https://github.com/mithrandie/csvq/>) and integrated to WA2L/edrc by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

csv2worksheet – load CSV file into Excel Worksheet

SYNOPSIS

edrc/lib/csv2worksheet [**-h**]

csv2worksheet [**-f filename**] [{ **-e** | **-n** } *sheet*] [**-d delimiter**] [**-o** [*col*] , [*row*]] [**-s cols**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

With the **csv2worksheet** command a CSV file (comma separated ASCII file) can be loaded into an existing Microsoft Excel (TM) **.xlsx** spreadsheet.

An Excel spreadsheet, also known as workbook, consists of one or more sheets, also known as worksheets. This worksheets are represented as named tabs in a workbook. With **csv2worksheet** it is possible to load a file that is structured into rows and the rows consist of fields separated by a delimiter into a worksheet. The row field separator defaults to ; but can be specified with the **-d** option.

This is an easy method to automatically create "nice looking" Excel reports based on plain ASCII data without the need to create Excel makros or to develop hard to maintain programs that create the whole Excel sheet on the fly. The idea is to load the data into a separate "data" worksheet and to perform the calculations, graphical data representation, statistics etc. in an other worksheet in the same Excel workbook which is not affected by the data load.

The advantage of this method is, that reports can be generated automatically without manual intervention (except the generation of the initial Excel sheet) even on non Windows systems, such as Unix or Linux.

It is possible to specify the name on an existing worksheet or to create a new worksheet while loading the data, dependent on your requirements.

However, the **csv2worksheet** command cannot overwrite existing data in a worksheet.

The origin of the data load can be specified. This does mean that data does not need to be loaded into a worksheet starting in cell A:1, a different cell can be specified. This enables the user to load multiple CSV data into the same worksheet as long as the data is loaded into ascending areas of an empty worksheet range.

OPTIONS

-h usage message.

-f filename

Excel file where the CSV data should be loaded into. This file has to be writeable by the user invoking **csv2worksheet**. Be aware, that **csv2worksheet** alters the data in the Excel file specified in the **-f** option and does not create a copy of the file. Therefore you should keep your

original template in a safe place. A good practice is to copy the original Excel file first and then to load the data. If you do not specify a file with the **-f filename** option, the file **io.xlsx** is used as a default.

-e existing_worksheet_name

load the CSV data into the existing worksheet with the name specified. If the worksheet does not exist in the workbook, **csv2worksheet** does exit without loading any data. If you neither specify a file with the **-e** nor the **-n** option, **csv2worksheet** tries to load the data into the existing **DATA** worksheet, that is therefore the default.

-n new_worksheet_name

load the CSV data into a worksheet with the name specified. If the worksheet does not exist currently, it is created at the end of the already existing worksheets.

-d delimiter

delimiter that separates the fields of a row. The delimiter is not restricted to a single character, but spaces or tabulators are not allowed. The default delimiter is **;** if this option is not specified.

-o column_number,row_number

column number and row number where to start the load of the CSV data into the worksheet. If this option is not specified, the data is loaded starting at column 1,1, that means the upper left corner of the data loaded is in cell A:1.

The following examples are allowed specifications: **-o 3,8** (start at cell C:8), **-o ,8** (start at cell A:8), **-o 3**, (start at cell C:1), **-o** , (start at cell A:1).

-s column_list

comma separated list of source column numbers that should be loaded explicitly as text. The normal behaviour of **csv2worksheet** is, that whenever a number is recognized in the input, it is loaded as a floating point value while all other data is loaded as text. See also **NOTES** section.

If you specify **-s 0** all columns will be loaded as text.

ENVIRONMENT

CSV2WORKSHEET_JAVA_OPTIONS="*option*"

additional options for the **Java** JVM process.

1) Example: set heap space

```
CSV2WORKSHEET_JAVA_OPTIONS="-Xmx8g"; export CSV2WORKSHEET_JAVA_OPTIONS
```

to set the **Java** heap space to 8 GBytes which might be needed when loading a large **csv** file into a worksheet and receiving the "java.lang.OutOfMemoryError: Java heap space" error.

To query the default JVM heap size use:

```
java -XX:+PrintFlagsFinal -version | grep HeapSize
```


2) Example: relocate temp files

```
CSV2WORKSHEET_JAVA_OPTIONS="-Djava.io.tmpdir=/dat/tmp"
export CSV2WORKSHEET_JAVA_OPTIONS
```

to relocate the temporary files generated by **csv2worksheet** from **/tmp/** to **/dat/tmp/**.

EXIT STATUS

| | |
|----------|--|
| 0 | no error. |
| 1 | the specified worksheet in the workbook could not be opened or created. |
| 2 | Excel file is not writable by the user invoking csv2worksheet or it does not exist. |
| 4 | usage displayed. |
| 5 | version displayed. |
| 6 | cannot load data. Ensure the worksheet range where to load the data is empty. |

FILES

edrc/var/csv2worksheet/io.xlsx

Empty Microsoft Excel workbook. This workbook contains the worksheets **Sheet1**, **Sheet2**, **Sheet3** and **DATA** and can be used as a template to load data into.

EXAMPLES

1) load data into an existing worksheet

Load file **data.csv** into the existing worksheet '*current*' of file **statistics.xlsx**

```
[ /data_dwh1/dat/exports ]
[ root@dwh_dbl_prod ][ksh]: cat data.csv | \
                           csv2worksheet -f statistics.xlsx \
                           -e current
```

csv2worksheet - load a CSV file into an Excel sheet, by Chr. Walther

```
load data ...
  sheet 'statistics.xlsx/current' opened.
  load origin is '1,1'.
  data delimiter is ';'.
  explicit text columns are ''.
  load rows ...
    1 ..... (36)
    2 ..... (36)
```

```

3      ..... (36)
4      ..... (36)
5      ..... (15)
6      ..... (36)
7      ..... (36)
8      ..... (36)
9      ..... (29)
10     ..... (36)
done.
done.

```

2) load data into a new worksheet using input redirection

Load file **data.csv** into the new worksheet '*Month=Sep*' of file **statistics.xlsx** starting in cell C:5 using 'input redirection' instead of a pipe

```

[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: csv2worksheet < data.csv \
                           -f statistics.xlsx \
                           -n Month='date +%b' -o 3,5

```

csv2worksheet - load a CSV file into an Excel sheet, by Chr. Walther

```

load data ...
  sheet 'statistics.xlsx/Month=Sep' created.
  load origin is '3,5'.
  data delimitator is ';'.
  explicit text columns are ''.
  load rows ...
5      ..... (36)
6      ..... (36)
7      ..... (36)
8      ..... (36)
9      ..... (15)
10     ..... (36)
11     ..... (36)
12     ..... (36)
13     ..... (29)
14     ..... (36)
done.
done.

```

3) load data into existing worksheet using in here

Load data into the existing worksheet '*DATA*' of file **io.xlsx** using the 'in here' mechanism

```

[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: cat <<EOM | csv2worksheet
NAME;MIN;MAX
alpha;100;200
bravo;150;250
charly;155;400
EOM

```

csv2worksheet - load a CSV file into an Excel sheet, by Chr. Walther

```
load data ...
  sheet 'io.xlsx/DATA' opened.
  load origin is '1,1'.
  data delimiter is ';'.
  explicit text columns are ''.
  load rows ...
    1      ... (3)
    2      ... (3)
    3      ... (3)
    4      ... (3)
  done.
done.
```

4) set a single cell of existing worksheet

Set a single cell (E:20) of an existing worksheet 'DATA' of file **io.xlsx**

```
[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: echo "Load at: `date`" | \
                           csv2worksheet -o 5,20
```

csv2worksheet - load a CSV file into an Excel sheet, by Chr. Walther

```
load data ...
  sheet 'io.xlsx/DATA' opened.
  load origin is '5,20'.
  data delimiter is ';'.
  explicit text columns are ''.
  load rows ...
    20      . (1)
  done.
done.
```

5) explicitly load certain columns as text

Load data into the existing worksheet 'DATA' of file io.xlsx using the 'in here' mechanism and explicitly load the columns 4 and 5 (of the input) as text into the worksheet.

```
[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: cat <<EOM | csv2worksheet -s 4,5
NAME;MIN;MAX;SERIAL;PHONE
alpha;100;200;2009063009250001;080007121941
bravo;150;250;2009063009250002;080006061944
charly;155;400;2009063009250003;080020071969
EOM
```

csv2worksheet - load a CSV file into an Excel sheet, by Chr. Walther

```
load data ...
  sheet 'io.xlsx/DATA' opened.
  load origin is '1,1'.
```

```

data delimiter is ';' .
explicit text columns are '4,5' .
load rows ...
    1      ..... (3)
    2      ..... (3)
    3      ..... (3)
    4      ..... (3)
done .
done .

```

SEE ALSO

edrcintro(1), Microsoft Excel Help

NOTES

If the field in the CSV file is recognized as a number, it is loaded into the worksheet as a "double" floating point value (+/-1.79769313486231570 * 10³⁰⁸). Otherwise it is loaded as a text (string) value.

csv2worksheet uses "**Jakarta POI – Java API To Access Microsoft Format Files**" of the Apache Jakarta project. See <http://jakarta.apache.org/poi/> for more information about the progress of the implementation.

BUGS

Beginning with version 1.1.11 of 2022, the **csv2worksheet** command does no longer support the since 2006 supported old ***.xls** file format. This for the sake of being now able to process very large worksheet files without memory and performance problems.

Therefore if still jobs are present that operate with ***.xls** files, convert this files to ***.xlsx**.

AUTHOR

csv2worksheet was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2006-2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

datalist.dat – datalist definition for lots

SYNOPSIS

edrc/var/lots/objects/datalist.dat
VARDIR/objects/datalist.dat

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the datalist definition for the **lots** command.

In this file sets of files are defined that can be scheduled to be saved in the **schedule.dat** file.

FILEFORMAT

The fileformat is a list of definitions that have the format

DATALIST ; DESCRIPTION ; DIRECTORY ;

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

It is allowed to have multiple lines in the file having the identical *DATALIST* definition, but different *DIRECTORY* entries to enable to save data from different directories with one schedule. If multiple lines are defined, ensure that all *DESCRIPTION* entries of those lines are identical due to the fact that only the first *DESCRIPTION* definition of a *DATALIST* will be saved in the **lots** job and conserved for the duration of the data retention.

OPTIONS*DATALIST*

name of the data list, the set of directories or files to be saved. To save data defined here in the *DATALIST*, *DIRECTORY* pair, a relating *DATALIST* entry in the **schedule.dat** file must exist.

DESCRIPTION

description of the datalist respectively of the data to be saved.

Be aware that this should be a meaningful description due to the fact that this data is saved for a long time and people having done the definition might have moved on before the data might be needed for restore.

DIRECTORY

definition of the directory or file(s) to be saved in this set. If a directory is specified, all files and subdirectories are saved. If a file is specified only the matching file(s) are saved. It is allowed to use wild-cards, but if you want to save multiple directories that cannot be defined using wild-cards, create multiple *DATALIST* entries.

To resolve the files/directories to be saved, **lots** uses

find DIRECTORY -print

internally. Therefore if you are unsure if your definition is correct, use this command for verification or start a **collect** run using '**lots -a collect -d DATALIST**' to verify the setting.

EXAMPLES

```
#
# var/lots/objects/datalist.dat - data save definition
#
# [00] 30.09.2009 CWa Initial Version
#
#DATALIST      ;DESCRIPTION                                ;DIRECTORY;
DB_CEDPROD     ;Database, PRODUCTION, PReCED              ;/orara/be/
DB_COIPROD     ;Database, PRODUCTION, OpenCO              ;/orara/be/
DB_DMSPROD     ;Database, PRODUCTION, Phoenix DocumentManagementSystem ;/orara/be/

DB_ASYPROD     ;Database, PRODUCTION, AutoSYS Job Scheduler ;/orara/be/

SQL_CHaT       ;MSSQL Database, PRODUCTION, CHaT          ;/dat/sqlra
SQL_plumdb     ;MSSQL Database, PRODUCTION, plumdb        ;/dat/sqlra
```

SEE ALSO

edrcintro(1), **schedule.dat(4)**, **volume.dat(4)**, **lots.cfg(4)**, **lots(1m)**

NOTES

To verify the datalist definitions, use the **lots -a list_datalist** command. Only correct entries will be listed.

BUGS

-

AUTHOR

datalist.dat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`daemon_wrapper` – wrap daemons written in Bourne shell

SYNOPSIS

`edrc/bin/daemon_command -> .daemon_wrapper`

AVAILABILITY

WA2L/edrc

DESCRIPTION

start commands that run as daemons out of the **edrc/lib/daemon** directory.

The *daemon_command* is started with the directory first changed to root (*/*) if **PWDPRESERVE=daemon_command** is not set in the *daemon_command*.

The **.daemon_wrapper** command is used for two purposes:

- 1) start a command from the root (*/*) working directory.
- 2) allow multiple versions of the same command to be present in the **lib/daemon/** directory without to pollute the **bin/** directory with all the symbolic links.

Typically commands that need to be present in different versions simultaneously are commands having long run times, daemons or commands for special purposes, where the program might run during a patch update of the package.

Examples for daemon type commands are: **logcheckd(1m)**, **job(1)** and **passwdsyncd(1m)**.

An example for a long running program is: **lots(1m)**.

An example for a special purpose program is: **patchinstall(1m)**.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- 101** Bourne shell not found in any of the directories: **/bin**, **/sbin** or **/usr/local/bin**.
- 102** program **edrc/lib/daemon/daemon_command** does not exist.
- 103** user calling the command has no permission to access/execute the called *daemon_command*.
- 107** direct call of **.daemon_wrapper** not allowed.

FILES

lib/daemon/
location of the commands started as daemons.

EXAMPLES

-

SEE ALSO

edrcintro(1), **cmdlist(1m)**, **patchinstall(1m)**

NOTES

The **.daemon_wrapper** command is one of the main reasons, why commands in the WA2L/edrc package do not have to be stopped during a patch installation.

BUGS

-

AUTHOR

daemon_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

days – print days since January 1st of a given date

SYNOPSIS

edrc/lib/days [**-h**]

days [*MM/DD/YYYY* | *YYYY-MM-DD*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Evaluate days since January 1st of a given date.

If no date is specified the number of days from January 1st until the current date is printed (this equals to **date +%j** on HP-UX systems).

OPTIONS

-h usage message.

MM/DD/YYYY
date in the format "Month/Day/Year".

YYYY-MM-DD
date in the format "Year/Month/Day".

ENVIRONMENT

\$TZ time zone.

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES**1) print days since January 1st 1971 for January 28th 1971**

```
[ /home/mzv7t0 ]  
[ mzv7t0@rh7mzv7t001 ][ksh]: days 01/28/1971  
  
028
```

2) print days since January 1st 1980 for March 1st 1980 (this is leap year)

```
[ /home/mzv7t0 ]  
[ mzv7t0@rh7mzv7t001 ][ksh]: days 03/01/1980  
  
061
```

3) print days since January 1st 1981 for March 1st 1980

```
[ /home/mzv7t0 ]  
[ mzv7t0@rh7mzv7t001 ][ksh]: days 03/01/1981  
  
060
```

SEE ALSO

edrcintro(1), date(1), is_weekend(3), seconds(3), timer(1)

NOTES

-

BUGS

-

AUTHOR

days was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

dbrep – run reports/queries against an Oracle database

SYNOPSIS

edrc/bin/dbrep [**-h**]

dbrep -l

dbrep [**-i** *oracle_sid*] [**-Y**] [**-n**] [**-r** *report*] [**-o** *option_list*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

execute reports/queries against an Oracle database.

The **dbrep** allows to run prepared SQL scripts with a simple interface against an Oracle database.

OPTIONS

-h usage message.

-l list all reports.

-i *oracle_sid*
Oracle instance (default is **\$ORACLE_SID**).

-n no report header output.

-Y answer "execute alter report" question with yes.

-r *report* report to generate (default is growth).

-o *option_list*
list of report parameters.

ENVIRONMENT

\$ORACLE_SID

Oracle SID to connect ot by default.

EXIT STATUS

| | |
|----------|--|
| 0 | no error. |
| 2 | operating system not supported. |
| 3 | could not resolve an ORACLE_HOME . |
| 4 | usage printed. |
| 5 | command aborted. |
| 6 | configuration file etc/dbrep.cfg not found. |

FILES

edrc/etc/dbrep.cfg
configuration file of **dbrep**.

edrc/etc/oratab
Oracle **oratab** file.

edrc/etc/tnsnames.ora
Oracle **tnsnames.ora** file.

edrc/var/log/dbrep.log
log file of **dbrep**.

edrc/lib/dbrep/reports/<report>.sql
prepared SQL script of the report/query.

edrc/lib/dbrep/reports/<report>.desc
prepared SQL script description- and option definition for the the report/query.

edrc/lib/dbrep/admin/<script>.sql
SQL scripts for admin/setup tasks needed in the Oracle database to be queried.

EXAMPLES

-

SEE ALSO

edrcintro(1), **dbrep.cfg**(4), **oratab**(4), **tnsnames.ora**(4)

NOTES

dbrep uses **sqlplus** from Oracle to perform the reports/queries.

BUGS

dbrep needs the Oracle client to be installed on the system.

AUTHOR

dbrep was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

dbrep.cfg – configuration file for dbrep

SYNOPSIS

edrc/etc/dbrep.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **dbrep** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**TOOLS_USER**

Database default EDS Tools username and password to connect to the database.

Example: **TOOLS_USER**="tools_eds/groove"

Default: **TOOLS_USER**="tools_eds/groove"

NLS_LANG

NLS settings

Example: **NLS_LANG**=AMERICAN_AMERICA.WE8ISO8859P1

Default: **NLS_DATE_FORMAT**=

NLS_DATE_FORMAT

NLS settings

Example: **NLS_DATE_FORMAT**='MON DD YYYY HH24:MI:SS'

Default: **NLS_DATE_FORMAT**=

ORACLE_HOME_SEARCH

Oracle Home search regex

Example: ORACLE_HOME_SEARCH='/ora/product/9*'

Default: ORACLE_HOME_SEARCH=

SEE ALSO

edrcintro(1), **dbrep(1)**,

NOTES

-

BUGS

-

AUTHOR

dbrep.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

diff – compare files line by line

SYNOPSIS

diff [*OPTION*]... *FILES*

DESCRIPTION

Compare *FILES* line by line.

Mandatory arguments to long options are mandatory for short options too.

--normal

output a normal diff (the default)

-q, --brief

report only when files differ

-s, --report-identical-files

report when two files are the same

-c, -C NUM, --context[=*NUM*]

output *NUM* (default 3) lines of copied context

-u, -U NUM, --unified[=*NUM*]

output *NUM* (default 3) lines of unified context

-e, --ed

output an ed script

-n, --rcs

output an RCS format diff

-y, --side-by-side

output in two columns

-W, --width=*NUM*

output at most *NUM* (default 130) print columns

--left-column

output only the left column of common lines

--suppress-common-lines

do not output common lines

-p, --show-c-function

show which C function each change is in

-F, --show-function-line=*RE*

show the most recent line matching *RE*

--label LABEL

use *LABEL* instead of file name and timestamp (can be repeated)

-t, --expand-tabs

expand tabs to spaces in output

-T, --initial-tab

make tabs line up by prepending a tab

--tabsize=*NUM*

tab stops every *NUM* (default 8) print columns

--suppress-blank-empty

suppress space or tab before empty output lines

-l, --paginate

pass output through 'pr' to paginate it

-r, --recursive
recursively compare any subdirectories found

--no-dereference
don't follow symbolic links

-N, --new-file
treat absent files as empty

--unidirectional-new-file
treat absent first files as empty

--ignore-file-name-case
ignore case when comparing file names

--no-ignore-file-name-case
consider case when comparing file names

-x, --exclude=*PAT*
exclude files that match *PAT*

-X, --exclude-from=*FILE*
exclude files that match any pattern in *FILE*

-S, --starting-file=*FILE*
start with *FILE* when comparing directories

--from-file=*FILE1*
compare *FILE1* to all operands; *FILE1* can be a directory

--to-file=*FILE2*
compare all operands to *FILE2*; *FILE2* can be a directory

-i, --ignore-case
ignore case differences in file contents

-E, --ignore-tab-expansion
ignore changes due to tab expansion

-Z, --ignore-trailing-space
ignore white space at line end

-b, --ignore-space-change
ignore changes in the amount of white space

-w, --ignore-all-space
ignore all white space

-B, --ignore-blank-lines
ignore changes where lines are all blank

-I, --ignore-matching-lines=*RE*
ignore changes where all lines match *RE*

-a, --text
treat all files as text

--strip-trailing-cr
strip trailing carriage return on input

-D, --ifdef=*NAME*
output merged file with '#ifdef *NAME*' diffs

--GTYPE-group-format=*GFMT*
format *GTYPE* input groups with *GFMT*

--line-format=LFMT
format all input lines with LFMT

--LTYPE-line-format=LFMT
format LTYPE input lines with LFMT

These format options provide fine-grained control over the output of diff, generalizing **-D/--ifdef**.

LTYPE is 'old', 'new', or 'unchanged'.
GTYPE is LTYPE or 'changed'.
GFMT (only) may contain:

%< lines from FILE1
%> lines from FILE2
%= lines common to FILE1 and FILE2
%[-][WIDTH][.PREC]){doxX}LETTER
printf-style spec for LETTER
LETTERS are as follows for new group, lower case for old group:

F first line number
L last line number
N number of lines = L-F+1
E F-1
M L+1
%(A=B?T:E)
if A equals B then T else E
LFMT (only) may contain:

%L contents of line
%l contents of line, excluding any trailing newline
%[-][WIDTH][.PREC]){doxX}n
printf-style spec for input line number
Both GFMT and LFMT may contain:

%% %
%c'C' the single character C
%c'\OOO'
the character with octal code OOO
C the character C (other characters represent themselves)

-d, --minimal
try hard to find a smaller set of changes

--horizon-lines=NUM
keep NUM lines of the common prefix and suffix

--speed-large-files
assume large files and many scattered small changes

--color[=WHEN]
colorize the output; WHEN can be 'never', 'always', or 'auto' (the default)

--palette=PALETTE

the colors to use when **--color** is active; PALETTE is a colon-separated list of terminfo capabilities

--help display this help and exit

-v, --version

output version information and exit

FILES are 'FILE1 FILE2' or 'DIR1 DIR2' or 'DIR FILE' or 'FILE DIR'. If **--from-file** or **--to-file** is given, there are no restrictions on FILE(s). If a FILE is '-', read standard input. Exit status is 0 if inputs are the same, 1 if different, 2 if trouble.

AUTHOR

Written by Paul Eggert, Mike Haertel, David Hayes, Richard Stallman, and Len Tower.

REPORTING BUGS

Report bugs to: bug-diffutils@gnu.org

GNU diffutils home page: <http://www.gnu.org/software/diffutils/>

General help using GNU software: <http://www.gnu.org/gethelp/>

COPYRIGHT

Copyright © 2017 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later [<http://gnu.org/licenses/gpl.html>](http://gnu.org/licenses/gpl.html).

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

[wdiff\(1\)](#), [cmp\(1\)](#), [diff3\(1\)](#), [sdiff\(1\)](#), [patch\(1\)](#)

The full documentation for **diff** is maintained as a Texinfo manual. If the **info** and **diff** programs are properly installed at your site, the command

info diff

should give you access to the complete manual.

NAME

dig – send domain name query packets to name servers

SYNOPSIS

dig [*@server*] *domain* [*<query-type>*] [*<query-class>*] [*+<query-option>*] [*–<dig-option>*] [*%comment*]

DESCRIPTION

Dig (domain information groper) is a flexible command line tool which can be used to gather information from the Domain Name System servers. *Dig* has two modes: simple interactive mode which makes a single query, and batch which executes a query for each in a list of several query lines. All query options are accessible from the command line.

The usual simple use of *dig* will take the form:

```
dig @server domain query-type query-class
```

where:

server may be either a domain name or a dot-notation Internet address. If this optional field is omitted, *dig* will attempt to use the default name server for your machine.

Note: If a domain name is specified, this will be resolved using the domain name system resolver (i.e., BIND). If your system does not support DNS, you may *have* to specify a dot-notation address. Alternatively, if there is a server at your disposal somewhere, all that is required is that */etc/resolv.conf* be present and indicate where the default name servers reside, so that *server* itself can be resolved. See *resolver(5)* for information on */etc/resolv.conf*. (WARNING: Changing */etc/resolv.conf* will affect the standard resolver library and potentially several programs which use it.) As an option, the user may set the environment variable LOCALRES to name a file which is to be used instead of */etc/resolv.conf* (LOCALRES is specific to the *dig* resolver and not referenced by the standard resolver). If the LOCALRES variable is not set or the file is not readable then */etc/resolv.conf* will be used.

domain is the domain name for which you are requesting information. See OPTIONS [-x] for convenient way to specify inverse address query.

query-type

is the type of information (DNS query type) that you are requesting. If omitted, the default is "a" (T_A = address). The following types are recognized:

| | | |
|-------|---------|--|
| a | T_A | network address |
| any | T_ANY | all/any information about specified domain |
| mx | T_MX | mail exchanger for the domain |
| ns | T_NS | name servers |
| soa | T_SOA | zone of authority record |
| hinfo | T_HINFO | host information |
| axfr | T_AXFR | zone transfer (must ask an authoritative server) |
| txt | T_TXT | arbitrary number of strings (not yet supported by BIND) |

(See RFC 1035 for the complete list.)

query-class

is the network class requested in the query. If omitted, the default is "in" (C_IN = Internet). The following classes are recognized:

| | | |
|-----|-------|---------------------------|
| in | C_IN | Internet class domain |
| any | C_ANY | all/any class information |

(See RFC 1035 for the complete list.)

Note: "Any" can be used to specify a class and/or a type of query. *Dig* will parse the first occurrence of "any" to mean query-type = T_ANY. To specify query-class = C_ANY you must either specify "any" twice, or set query-class using "-c" option (see below).

OTHER OPTIONS

%ignored-comment

"%" is used to included an argument that is simply not parsed. This may be useful if running *dig* in batch mode. Instead of resolving every @server-domain-name in a list of queries, you can avoid the overhead of doing so, and still have the domain name on the command line as a reference. Example:

```
dig @128.9.0.32 %venera.isi.edu mx isis.edu
```

—<dig option>

"—" is used to specify an option which effects the operation of *dig*. The following options are currently available (although not guaranteed to be useful):

—x *dot-notation-address*

Convenient form to specify inverse address mapping. Instead of "dig 32.0.9.128.in-addr.arpa" one can simply "dig -x 128.9.0.32".

—f *file* File for *dig* batch mode. The file contains a list of query specifications (*dig* command lines) which are to be executed successively. Lines beginning with ';', '#', or '\n' are ignored. Other options may still appear on command line, and will be in effect for each batch query.

—T *time*

Time in seconds between start of successive queries when running in batch mode. Can be used to keep two or more batch *dig* commands running roughly in sync. Default is zero.

—p *port* Port number. Query a name server listening to a non-standard port number. Default is 53.

—P[*ping-string*]

After query returns, execute a *ping*(8) command for response time comparison. This rather unelegantly makes a call to the shell. The last three lines of statistics is printed for the command:

```
ping -s server_name 56 3
```

If the optional "ping string" is present, it replaces "ping -s" in the shell command.

—t *query-type*

Specify type of query. May specify either an integer value to be included in the type field or use the abbreviated mnemonic as discussed above (i.e., mx = T_MX).

—c *query-class*

Specify class of query. May specify either an integer value to be included in the class field or use the abbreviated mnemonic as discussed above (i.e., in = C_IN).

—envsav

This flag specifies that the *dig* environment (defaults, print options, etc.), after all of the arguments are parsed, should be saved to a file to become the default environment. Useful if you do not like the standard set of defaults and do not desire to include a large number of options each time *dig* is used. The environment consists of resolver state variable flags, timeout, and retries as well as the flags detailing *dig* output (see below). If the shell environment variable LOCALDEF is set to the name of a file, this is where the default *dig* environment is saved. If not, the file "DiG.env" is created in the current working

directory.

Note: LOCALDEF is specific to the *dig* resolver, and will not affect operation of the standard resolver library.

Each time *dig* is executed, it looks for *./DiG.env* or the file specified by the shell environment variable LOCALDEF. If such file exists and is readable, then the environment is restored from this file before any arguments are parsed.

-envset This flag only affects batch query runs. When *"-envset"* is specified on a line in a *dig* batch file, the *dig* environment after the arguments are parsed, becomes the default environment for the duration of the batch file, or until the next line which specifies *"-envset"*.

-[no]stick

This flag only affects batch query runs. It specifies that the *dig* environment (as read initially or set by *"-envset"* switch) is to be restored before each query (line) in a *dig* batch file. The default *"-nostick"* means that the *dig* environment does not stick, hence options specified on a single line in a *dig* batch file will remain in effect for subsequent lines (i.e. they are not restored to the "sticky" default).

+<query option>

"+" is used to specify an option to be changed in the query packet or to change *dig* output specifics. Many of these are the same parameters accepted by *nslookup*(1). If an option requires a parameter, the form is as follows:

+keyword[=value]

Most keywords can be abbreviated. Parsing of the *"+"* options is very simplistic — a value must not be separated from its keyword by white space. The following keywords are currently available:

| Keyword | Abbrev. | Meaning [default] |
|-------------|---------|---|
| [no]debug | (deb) | turn on/off debugging mode [deb] |
| [no]d2 | | turn on/off extra debugging mode [nod2] |
| [no]recurse | (rec) | use/don't use recursive lookup [rec] |
| retry=# | (ret) | set number of retries to # [4] |
| time=# | (ti) | set timeout length to # seconds [4] |
| [no]ko | | keep open option (implies vc) [noko] |
| [no]vc | | use/don't use virtual circuit [novc] |
| [no]defname | (def) | use/don't use default domain name [def] |
| [no]search | (sea) | use/don't use domain search list [sea] |
| domain=NAME | (do) | set default domain name to NAME |
| [no]ignore | (i) | ignore/don't ignore trunc. errors [noi] |
| [no]primary | (pr) | use/don't use primary server [nopr] |
| [no]aaonly | (aa) | authoritative query only flag [noaa] |
| [no]sort | (sor) | sort resource records [nosor] |
| [no]cmd | | echo parsed arguments [cmd] |
| [no]stats | (st) | print query statistics (RTT,etc) [st] |
| [no]Header | (H) | print basic header [H] |
| [no]header | (he) | print header flags [he] |
| [no]ttlid | (tt) | print TTLs [tt] |
| [no]cl | | print class info [nocl] |
| [no]qr | | print outgoing query [noqr] |
| [no]reply | (rep) | print reply [rep] |
| [no]ques | (qu) | print question section [qu] |

| | | |
|------------|------|--|
| [no]answer | (an) | print answer section [an] |
| [no]author | (au) | print authoritative section [au] |
| [no]addit | (ad) | print additional section [ad] |
| pfdef | | set to default print flags |
| pfmin | | set to minimal default print flags |
| pfset=# | | set print flags to # (# can be hex/octal/decimal) |
| pfand=# | | bitwise and print flags with # |
| pfor=# | | bitwise or print flags with # |

The retry and time options affect the retransmission strategy used by resolver library when sending datagram queries. The algorithm is as follows:

```

for i = 0 to retry - 1
  for j = 1 to num_servers
    send_query
    wait((time * (2**i)) / num_servers)
  end
end

```

(Note: *dig* always uses a value of 1 for num_servers.)

Pfset, pfand, and pfor were included to make manipulation of the various print options less tedious. Below are the currently defined meanings for the various print flag bits. They may be combined (ANDed) to achieve various output formats.

| | | |
|-----------|--------|--|
| PRF_STATS | 0x0001 | RTT, query & server host, date, msg size |
| PRF_CLASS | 0x0004 | Resource record class information |
| PRF_CMD | 0x0008 | dig command line echo |
| PRF_QUES | 0x0010 | questions section |
| PRF_ANS | 0x0020 | answers section |
| PRF_AUTH | 0x0040 | authoritative section |
| PRF_ADD | 0x0080 | additional records section |
| PRF_HEAD1 | 0x0100 | RR section headers & counts |
| PRF_HEAD2 | 0x0200 | pkt header flags |
| PRF_TTLID | 0x0400 | Resource record ttl |
| PRF_HEADX | 0x0800 | basic header |
| PRF_QUERY | 0x1000 | outgoing query packet |
| PRF_REPLY | 0x2000 | reply packet |
| PRF_SORT | 0x8000 | sort various response sections |
| PRF_DEF | 0x2ff9 | default dig settings |
| PRF_ZONE | 0x24f9 | default setting for zone transfer |
| PRF_MIN | 0xa930 | minimalistic dig settings for (future) automated server testing |

When setting the print options, if you want to see information other than statistics, you should choose to examine the outgoing (0x1000), incoming (0x2000), or both packets plus the specific sections of the packet you are interested in.

DETAILS

Dig requires a slightly modified version of the BIND *resolver*(3) library to gather count and time statistics. Otherwise, it is straight-forward (albeit not pretty) effort of parsing arguments and setting appropriate parameters. *Dig* uses resolver routines *res_init()*, *res_mkquery()*, *res_send()* as well as accessing *_res* structure. Compiling *dig* with the standard resolver library is possible, but will change the output format, make the print options meaningless, and not gather RTT and packet count stats.

FILES

/etc/resolv.conf initial domain name and name server
 addresses

ENVIRONMENT

LOCALRES file to use in place of /etc/resolv.conf
LOCALDEF default environment file

AUTHOR

Steve Hotz hotz@isi.edu

ACKNOWLEDGMENTS

Dig uses functions from *nslookup*(1) authored by Andrew Cherenon; taken from Berkeley BIND 4.8 distribution. The resolver library is primarily from the Berkeley 4.8 BIND distribution.

BUGS

Dig has a serious case of "creeping featurism" -- the result of considering several potential uses during it's development. It would probably benefit from a rigorous diet. Similarly, the print flags and granularity of the items they specify make evident their rather ad hoc genesis.

Dig does not consistently exit nicely (with appropriate status) when a problem occurs somewhere in the resolver (NOTE: most of the common exit cases are handled). This is particularly annoying when running in batch mode. If it exits abnormally (and is not caught), the entire batch aborts; when such an event is trapped, *dig* simply continues with the next query.

SEE ALSO

named(8), resolver(3), resolver(5), nslookup(1)

NAME

directories – filter to expand filelist with overlying directories

SYNOPSIS

edrc/lib/directories

AVAILABILITY

WA2L/edrc

DESCRIPTION

expand a file list data stream received via **stdin** with the overlying directories and print the result to **stdout**.

This can be used to ensure when copying data that the directory permissions are set correctly.

Suppose the following command is issued as the root user:

```
find /data/source/myApp -name *.java -print |\  
cpio -pdvum /backup/myApp/20100117
```

This will copy all ***.java** files in all underlying directories from **/data/source/myApp** to the **/backup/myApp/20100117** directory, but all directories will have the ownership of the root user and the permissions will be set accordingly to the umask of the root user. All ***.java** files will have the correct permissions and ownership.

When using the **directories** command as the following example:

```
find /data/source/myApp -name *.java -print | directories |\  
cpio -pdvum /backup/myApp/20100117
```

all permissions and all ownership will be correct.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **cpio(1)**, **find(1)**, **sparse(3)**

NOTES

-

BUGS

-

AUTHOR

directories was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

dmidecode – DMI table decoder

SYNOPSIS

dmidecode [**OPTIONS**]

DESCRIPTION

dmidecode is a tool for dumping a computer's DMI (some say SMBIOS) table contents in a human-readable format. This table contains a description of the system's hardware components, as well as other useful pieces of information such as serial numbers and BIOS revision. Thanks to this table, you can retrieve this information without having to probe for the actual hardware. While this is a good point in terms of report speed and safeness, this also makes the presented information possibly unreliable.

The DMI table doesn't only describe what the system is currently made of, it also can report the possible evolutions (such as the fastest supported CPU or the maximal amount of memory supported).

SMBIOS stands for System Management BIOS, while DMI stands for Desktop Management Interface. Both standards are tightly related and developed by the DMTF (Desktop Management Task Force).

As you run it, **dmidecode** will try to locate the DMI table. If it succeeds, it will then parse this table and display a list of records like this one:

```
Handle 0x0002, DMI type 2, 8 bytes. Base Board Information
Manufacturer: Intel
Product Name: C440GX+
Version: 727281-001
Serial Number: INCY92700942
```

Each record has:

- A handle. This is a unique identifier, which allows records to reference each other. For example, processor records usually reference cache memory records using their handles.
- A type. The SMBIOS specification defines different types of elements a computer can be made of. In this example, the type is 2, which means that the record contains "Base Board Information".
- A size. Each record has a 4-byte header (2 for the handle, 1 for the type, 1 for the size), the rest is used by the record data. This value doesn't take text strings into account (these are placed at the end of the record), so the actual length of the record may be (and is often) greater than the displayed value.
- Decoded values. The information presented of course depends on the type of record. Here, we learn about the board's manufacturer, model, version and serial number.

OPTIONS

-d, --dev-mem FILE

Read memory from device **FILE** (default: **/dev/mem**)

-q, --quiet

Be less verbose. Unknown, inactive and OEM-specific entries are not displayed. Meta-data and handle references are hidden. Mutually exclusive with **--dump**.

-s, --string KEYWORD

Only display the value of the DMI string identified by **KEYWORD**. **KEYWORD** must be a keyword from the following list: **bios-vendor**, **bios-version**, **bios-release-date**, **system-manufacturer**, **system-product-name**, **system-version**, **system-serial-number**, **system-uuid**, **baseboard-manufacturer**, **baseboard-product-name**, **baseboard-version**, **baseboard-serial-number**, **baseboard-asset-tag**, **chassis-manufacturer**, **chassis-type**, **chassis-version**, **chassis-serial-number**, **chassis-asset-tag**, **processor-family**, **processor-manufacturer**, **processor-version**,

processor-frequency. Each keyword corresponds to a given DMI type and a given offset within this entry type. Not all strings may be meaningful or even defined on all systems. Some keywords may return more than one result on some systems (e.g. **processor-version** on a multi-processor system). If **KEYWORD** is not provided or not valid, a list of all valid keywords is printed and **dmidecode** exits with an error. This option cannot be used more than once, and implies **--quiet**. Mutually exclusive with **--type** and **--dump**.

-t, --type TYPE

Only display the entries of type **TYPE**. **TYPE** can be either a DMI type number, or a comma-separated list of type numbers, or a keyword from the following list: **bios**, **system**, **baseboard**, **chassis**, **processor**, **memory**, **cache**, **connector**, **slot**. Refer to the DMI TYPES section below for details. If this option is used more than once, the set of displayed entries will be the union of all the given types. If **TYPE** is not provided or not valid, a list of all valid keywords is printed and **dmidecode** exits with an error. Mutually exclusive with **--string**.

-u, --dump

Do not decode the entries, dump their contents as hexadecimal instead. Note that this is still a text output, no binary data will be thrown upon you. The strings attached to each entry are displayed as both hexadecimal and ASCII. This option is mainly useful for debugging. Mutually exclusive with **--quiet** and **--string**.

-h, --help

Display usage information and exit

-V, --version

Display the version and exit

DMI TYPES

The SMBIOS specification defines the following DMI types:

| Type | Information |
|------|------------------------------|
| 0 | BIOS |
| 1 | System |
| 2 | Base Board |
| 3 | Chassis |
| 4 | Processor |
| 5 | Memory Controller |
| 6 | Memory Module |
| 7 | Cache |
| 8 | Port Connector |
| 9 | System Slots |
| 10 | On Board Devices |
| 11 | OEM Strings |
| 12 | System Configuration Options |
| 13 | BIOS Language |
| 14 | Group Associations |
| 15 | System Event Log |
| 16 | Physical Memory Array |
| 17 | Memory Device |
| 18 | 32-bit Memory Error |
| 19 | Memory Array Mapped Address |
| 20 | Memory Device Mapped Address |
| 21 | Built-in Pointing Device |
| 22 | Portable Battery |
| 23 | System Reset |

| | |
|----|----------------------------------|
| 24 | Hardware Security |
| 25 | System Power Controls |
| 26 | Voltage Probe |
| 27 | Cooling Device |
| 28 | Temperature Probe |
| 29 | Electrical Current Probe |
| 30 | Out-of-band Remote Access |
| 31 | Boot Integrity Services |
| 32 | System Boot |
| 33 | 64-bit Memory Error |
| 34 | Management Device |
| 35 | Management Device Component |
| 36 | Management Device Threshold Data |
| 37 | Memory Channel |
| 38 | IPMI Device |
| 39 | Power Supply |

Additionally, type 126 is used for disabled entries and type 127 is an end-of-table marker. Types 128 to 255 are for OEM-specific data. **dmidecode** will display these entries by default, but it can only decode them when the vendors have contributed documentation or code for them.

Keywords can be used instead of type numbers with **--type**. Each keyword is equivalent to a list of type numbers:

| Keyword | Types |
|-----------|-------------------|
| bios | 0, 13 |
| system | 1, 12, 15, 23, 32 |
| baseboard | 2, 10 |
| chassis | 3 |
| processor | 4 |
| memory | 5, 6, 16, 17 |
| cache | 7 |
| connector | 8 |
| slot | 9 |

Keywords are matched case-insensitively. The following command lines are equivalent:

- `dmidecode --type 0 --type 13`
- `dmidecode --type 0,13`
- `dmidecode --type bios`
- `dmidecode --type BIOS`

FILES

`/dev/mem`

BUGS

More often than not, information contained in the DMI tables is inaccurate, incomplete or simply wrong.

AUTHORS

Alan Cox, Jean Delvare

SEE ALSO

biosdecode(8), **mem(4)**, **ownership(8)**, **vpddecode(8)**

NAME

dos2ux – convert ASCII file format from DOS to UNIX format

SYNOPSIS

edrc/lib/dos2ux [**-h**]

dos2ux *dosfile* > *unixfile*

cat *dosfile* | **dos2ux** > *unixfile*

AVAILABILITY

WA2L/edrc

DESCRIPTION

dos2ux reads the specified *dosfile* and writes it to **stdout**, converting to UNIX format.

If no input file is given or if a *unixfile* is specified as -, **dos2ux** reads from **stdin**.

OPTIONS

-h print usage message.

unixfile file in UNIX file format.

dosfile file in DOS file format.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 given file does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ux2dos(3)**

NOTES

-

BUGS

-

AUTHOR

dos2ux was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

..du_index – indexfile/description for du_report

SYNOPSIS

..du_index

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the "disk usage index" file created/edited by **duvi** and resolved by **du_report**.

This file has to be owned by 'root' and has to have the permissions '700' to be resolved by **du_report**.

FILEFORMAT

Plain ASCII file:

```
#
# ..du_index - indexfile/description for du_report
#
# [00] 20.12.2005 root          Initial Version
#
de: <Beschreibung in Deutsch>
en: <description text in English>
```

SEE ALSO

edrcintro(1), **du_report(1)**, **duvi(1)**

NOTES

-

BUGS

-

AUTHOR

..du_index was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

du_report – create a disk usage summary report

SYNOPSIS

edrc/bin/du_report [-h]

du_report [-c][-s][-l *language*][-f *output_file*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Create a disk usage summary report based on the directories labeled with "disk usage index" files (see: **du_index**(4)) using the **duvi**(1) command.

du_report searches **..du_index** files to create a report which lists the purpose of all labeled directories and some additional information as the size of the directory in kByte.

du_report only includes "disk usage index" files that have the correct file permissions of '700'.

OPTIONS

-h usage message.

-l *language*
generate the report in the corresponding *language*. The language is resolved from the index file.

-c use index cache file to create the report. The cache file contains all "disk usage index" files of previous runs. When you know, that no additional index files have been created since the last run of **du_report** in the root (/) directory, the report creation is speed up drastically due to the fact that not the whole system has to be searched for index files. All index files in the cache are verified against the current situation ahead of do the reporting. If an index file has been created in a certain sub directory (e.g. /export/dat/development/docs) and **du_report has been called while staying in this directory, it is added** to the index cache file, too.

-s do not summarize the disk usage for the directories.

-f *output_file*
file where to save the report to. If this option is not specified, the report is printed to stdout.

ENVIRONMENT

-

EXIT STATUS

- | | |
|-----------|--|
| 0 | no error. |
| 1 | du_report has been started as another user then 'root'. |
| 1 | the directory where to place the <i>output_file</i> does not exist. |
| 4 | usage printed. |
| 5 | report creation aborted. |
| 11 | temporary directory could not be claimed or created in /tmp . Check the system temporary directory /tmp if you get this error, it is an indicator of system intrusion. |

FILES

..du_index "disk usage" index file.

/var/tmp/du_report.indexfilecache

cache of all "disk usage index" files found during previous runs. If **du_report** is called using the **-c** option, this file is read.

EXAMPLES

-

SEE ALSO

duvi(1), **du_index(4)**, **edrcintro(1)**, **vi(1)**

NOTES

-

BUGS

-

AUTHOR

du_report was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

duvi – edit/create the disk usage index file

SYNOPSIS

edrc/bin/duvi [**-h**]

duvi [**-l** *language*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Create or edit the "disk usage index" file. This file (**..du_index**) contains the purpose (usage) of a certain directory. It is like a label of a directory.

After wards this file is searched by the **du_report** command to create a report which lists the usage of all labeled directories with and some additional information like the size of the directory in kByte.

duvi has to be started as user 'root' and saves the **..du_index** file with the access permissions of '700'. **du_report** will only include files with those credentials into the report.

To label a directory you have to change to the directory and call the **duvi** command. This command loads an existing **..du_index** file into **vi** or provides a template to be filled out. The **..du_index** file can hold the description of the directory in different languages.

The provided languages are:

en for English

de for german

With this mechanism it is very easy to document a system without maintaining extra lists which document the purpose of certain directories.

OPTIONS

-h usage message.

-l *language*
 jump to the corresponding *language* : token in the **..du_index** file.

FILES

..du_index "disk usage" index file.

etc/exrc settings for the **vi** / **vim** editor.

SEE ALSO

edrcintro(1), **du_report(1)**, **du_index(4)**, **vi(1)**, **swvi(1)**

NOTES

To see how to efficiently maintain a software inventory, see **swvi(1)**.

BUGS

-

AUTHOR

duvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ecrontab – maintains crontab files for individual users (cron in WA2L/edrc)

SYNOPSIS

edrc/bin/ecrontab [-u *user*] <*file* | ->

ecrontab [-T] <*file* | ->

ecrontab [-u *user*] <-l | -r | -e> [-i] [-s]

ecrontab -V

DESCRIPTION

ecrontab is the program used to install a crontab table *file*, remove or list the existing tables used to serve the **crond**(3) daemon. Each user can have their own crontab, and though these are files in **edrc/var/spool/cron/**, they are not intended to be edited directly.

Prior to editing a crontab table it is automatically saved using **vsav**(1).

Scheduling cron jobs with **ecrontab** can be allowed or disallowed for different users. For this purpose, use the **edrc/etc/cron.allow** and **edrc/etc/cron.deny** files. If the **cron.allow** file exists, a user must be listed in it to be allowed to use **ecrontab**. If the **cron.allow** file does not exist but the **cron.deny** file does exist, then a user must *not* be listed in the **cron.deny** file in order to use **ecrontab**. If neither of these files exist, then only the super user is allowed to use **ecrontab**.

The temporary directory can be set in an environment variable. If it is not set by the user, the **/tmp/** directory is used.

When listing a crontab on a terminal the output will be colorized unless an environment variable **NO_COLOR** is set.

OPTIONS

- u** Specifies the name of the user whose crontab is to be modified. If this option is not used, **ecrontab** examines "your" crontab, i.e., the crontab of the person executing the command. If no crontab exists for a particular user, it is created for them the first time the **ecrontab -u** command is used under their username.
- T** Test the crontab file syntax without installing it. Once an issue is found, the validation is interrupted, so this will not return all the existing issues at the same execution.
- l** Displays the current crontab on standard output.
- r** Removes the current crontab.
- e** Edits the current crontab using the editor specified by the **VISUAL** or **EDITOR** environment variables. After you exit from the editor, the modified crontab will be installed automatically.

Be aware, that the configuration file is saved automatically before editing using the **vsav**(1) command internally. Therefore when a messed-up file was saved, or to check changes, the **vls**(1), **vdiff**(1), **vrestore**(1) etc. commands can be used to verify changes or revert to a previous version.

- i** This option modifies the **-r** option to prompt the user for a 'y/Y' response before actually removing the crontab.

- s** Appends the current SELinux security context string as an **MLS_LEVEL** setting to the crontab file before editing / replacement occurs - see the documentation of **MLS_LEVEL** in **crontab(5)**.
- V** Print version and exit.

CAVEATS

The files **cron.allow** and **cron.deny** cannot be used to restrict the execution of cron jobs; they only restrict the use of **ecrontab**. In particular, restricting access to **ecrontab** has no effect on an existing **crontab** of a user. Its jobs will continue to be executed until the crontab is removed.

The files **cron.allow** and **cron.deny** must be readable by the user invoking **crontab**. If this is not the case, then they are treated as non-existent.

SEE ALSO

edrcintro(1), **crontab(4)**, **crond(3)**, **ecronnext(1)**, **vdiff(1)**, **vls(1)**, **vrestore(1)**, **vsav(1)**

FILES

edrc/etc/cron.allow

List of users to allow **ecrontab** access.

edrc/etc/cron.deny

List of users to deny **ecrontab** access.

STANDARDS

The **ecrontab** command conforms to IEEE Std1003.2-1992 (“POSIX”) with one exception: For replacing the current crontab with data from standard input the **-** has to be specified on the command line if the standard input is a TTY. This new command syntax differs from previous versions of Vixie Cron, as well as from the classic SVR3 syntax.

DIAGNOSTICS

An informative usage message appears if you run a crontab with a faulty command defined in it.

AUTHOR

Paul Vixie <vixie@isc.org>

Colin Dean <colin@colin-dean.org>

NAME

ecronnext – time of next job cron will execute (cron in WA2L/edrc)

SYNOPSIS

edrc/bin/ecronnext

[-i users] [-e users] [-s] [-a] [-t time] [-q time] [-j command] [-l] [-c] [-f] [-h] [-V] [file]...

DESCRIPTION

Determine the time **crond** will execute the next job. Without arguments, it prints that time considering all crontabs, in number of seconds since the Epoch, rounded to the minute. This number can be converted into other formats using **date(1)**, like **date --date @43243254**

The file arguments are optional. If provided, *ecronnext* uses them as crontabs instead of the ones installed in the system.

OPTIONS

-i user,user,user,...

Consider only the crontabs of the specified users. Use ***system*** for the system crontab.

-e user,user,user,...

Do not consider the crontabs of the specified users.

-s

Do not consider the system crontab, usually the **edrc/etc/crontab** file. The system crontab usually contains the hourly, daily, weekly and montly crontabs.

-a

Use the crontabs installed in the system in addition to the ones passed as file arguments. This is implicit if no file is passed.

-t time

Determine the next job from this time, instead of now. The time is expressed in number of seconds since the Epoch, as obtained for example by **date +%s --date "now + 2 hours"**, and is internally rounded to the minute.

-q time

Do not check jobs over this time, expressed in the same way as in option **-t**.

-j command

Only look for jobs that contain *command* as a substring.

-l

Print the whole entries of the jobs that are the next to be executed by **crond(3)**. The default is to only print their next time of execution.

-c

Print every entry in every crontab with the next time it is executed.

-f

Print all jobs that are executed in the given interval. Requires option **-q**.

-h

Print usage output and exit.

-V

Print version and exit.

AUTHOR

Marco Migliori <sgerwk@aol.com>

SEE ALSO

edrcintro(1), crond(3), crontab(4), date(1), ecrontab(1)

NAME

edrc – EDRC, Enterprise Disaster Recovery Console

SYNOPSIS

edrc/sbin/edrc [**-h** | **-V**]

edrc [**-c** *cfg*] [**-n** *session_name*] [**-s**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **edrc** program is the core part of the WA2L/edrc environment. The EDRC provides an easy to use, easy to customize and easy to enhance framework to recover your environment after a "Disaster Case".

The EDRC simplify the development of recovery scripts and helps to recover your environment in an efficient and reliable way.

The WA2L/edrc package has to be an integral part of a whole disaster recovery plan and process. This process is documented step by step in an "Emergency Handbook".

edrc is an interactive (character oriented) tool that supports the administrator and gives the freedom to change settings and invoked scripts at any point.

OPTIONS

-h usage message.

-V print version and patch level of **edrc**. For an explanation of the release numbering system see **edrcrevision**(1).

-c *configfile*
specify a special configuration file. If you do not use this option the default configuration file **edrc/etc/edrc.cfg** is used.

-n *session_name*
specify the session name on the command line. If you use this option you cannot set and alter the session name with the **name** EDRC command during runtime. Therefore the session name is read only if you use the **-n** option.

This can be used to ensure that only one **edrc** instance of a certain configuration is started per host (e.g.: **-n JOBCTL**).

If a session has to be marked, but it is allowed to start multiple instances of the same configuration, the **@ID@** field should be used in the **-n** option to generate a dynamic read only session name. The **@ID@** field is replaced by the process id of the started **edrc** instance, but if there is already an instance running with the same name it is replaced by a random number (e.g.: **-n JOB_@ID@**).

When session names are passed thru trunks to remote systems the remote session name is set equal to the local session name if possible.

If there is already a session with the same name running and the **@ID@** field is used as part of the session name, the **@ID@** field content will be resolved new and the trunk can be established. If the **@ID@** field is not used the trunk will not be established due to the fact that only one session with the identical name is allowed to be set/started per host.

Hint: when starting **sat(1)** the option **-n SAT_@ID@** is passed internally to **edrc**.

- s** silent startup. startup without showing the EDRC banner.
- t** no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

COMMANDS

This section describes all interactive internal commands of **edrc**;

- !<number>** execute the command with the number *<number>* listed in the **edrc** history. To list the history use the **history** command.
- !!** execute the command you invoked the last. This command is evaluated from the **edrc** history. To list the history use the **history** command.
- @** execute the menupoint you invoked the last. This command re-invokes the menupoint which is displayed after the ":" in the EDRC prompt.
- b, back, up** go one menu back. Respectively exit a submenu and go back to the overlying menu.
- banner** display the program startup banner with the copyright and version information.
- busy** print a rotating slash. This command does nothing productive. It might be useful to keep a line open when for instance a firewall cuts the connection if a TCP timeout is reached. To end the rotating press Ctrl+C.
- clear** clear screen.
- date** print the current system date and time.

debug toggle the script debug variable **\$DEBUG** between *False* and *True*. All recovery shells scripts should have the following line included to be able to react to the **debug** command:

```
test "$DEBUG" = True && set -x
```

After startup the **\$DEBUG** variable is set to *False*.

distribute distribute recovery scripts to other hosts. This command allows you to keep all recovery scripts on all your hosts synchronized. It is recommended that you set up your recovery menu tree that way that you can have an identical set of scripts on all your hosts. This has the advantage of a "single point of administration".

See **edrc.cfg(4)** for information about the configuration and the configuration file names.

The **distribute** command will guide you to a couple of steps needed to transfer the scripts to a set of hosts. It is also possible to enter hostgroups (*@HOSTGROUP*), as known from the **hostlist** command.

You will have the opportunity to cancel at several points.

Distribution types:

To speed up distribution and to allow some parallel editing on different hosts you can choose between a *differential* and a *full* distribution. A *full* distribution contains all files in a recovery script tree, a *differential* distribution contains only the files that are newer then the last *full* distribution.

The following guideline helps to decide which distribution type to use in what situation: If things are removed or renamed, use a *full* distribution, in all other cases use a *differential* distribution.

The following steps are performed by the distribute command:

- 1.) create a new distribution file. This file contains scripts in **SCRIPTS_BASEDIR**. If you decide to create a *full* distribution, all files will be packed. If you decide to create a *differential* distribution, only the files will be packed which are newer then the last distribution file created by a *full* distribution.
- 2.) specify a list of hosts to distribute the distribution file to.
- 3.) distribute (copy) the distribution file to each target host in the list.
- 4.) on the target host backup the current scripts. If it is a *full* distribution all scripts are backed up. If it is a *differential* distribution only the files which will be overwritten are backed up.
- 5.) if it is a full distribution remove all scripts in **SCRIPTS_BASEDIR** on the target host.
- 6.) extract the distribution file to the **SCRIPTS_BASEDIR** on the target host.

Hint:

The host where you invoke the **distribute** command is excluded from distribution. Therefore you can keep the local host in the **DIST_HOSTLIST** and the configuration file **edrc.cfg** can be kept identical on all your hosts.

To view the contents of a compressed backup file (without decompressing it) you can use the **lscomp**, **llcomp** and **catcomp** commands in the shell.

edrcdebug this will toggle debugging on the **edrc** command between *False* and *True*. Normally only the author of **edrc** will use this command for debugging of the EDRC itself. To debug your recovery scripts use the **debug** command.

edrcperm list the complete permission set and all functionality that is defined to be denied. The denial of the startup of certain functions can be specified in the **edrc.cfg** file with the **DENY_LIST** setting.

For a disaster recovery configuration, you should **never** deny any functionality from startup, due to the fact that in a disaster recovery maximal flexibility has to be provided under all circumstances.

Therefore the feature to deny certain functions from execution should only be used for system administration configurations, and even there - use it moderately! The user gets the message: "edrc-WARNING: execution of functionality '<option>' not permitted." if a certain command/functionality is not permitted to be executed. See also **edrc.cfg**(4) for more information.

env create, edit, view, remove the environment file. The environment file (**_env**) is used to do common settings for a certain menu (=directory).

So you can set passwords, ORACLE_SID's etc. in this file and you don't have to edit all your recovery scripts in this menu if you change the password (for instance). You could also do some user prompting in this environment file if you have to.

To keep the process and the dependencies simple the environment settings are not inherited to sub-menus.

If you like to share a whole recovery script tree later with other system administrators you can strip out all passwords from **_env** files with the **envpasswdstrip** shell command. This command triggers for patterns like 'export*password=my_secret' and 'export*passwd=my_secret' and substitutes the password with the string 'PASSWORD_STRIPED_OUT'.

How a recovery script is started:

- 1.) If an environment file (**_env**) exists for the current menu (=directory), the **_env** file is sourced and the variables defined in it are set. Your code has to be ksh conform and the variables have to be exported.
- 2.) If the recovery script is executable, invoke it directly. It is also validated, if the command defined in the recovery scripts magic key exists. If the shell in the magic key does not exist a fallback mechanism tries to resolve an alternate shell to be used to start the recovery script.
- 3.) If the recovery script is not executable, start it in using the shell as defined in the magic key. Here the fallback mechanism as described above applies as well.

Hint:

To be sure that your settings will reach the recovery script you should always export your variables.

When you create a new environment file a template is provided.

The environment file has the filepermissions 600.

Example _env file:

```
#
# _env - Environment settings for commands in ....
#
# [00] 29.01.2003 CWa Initial Version
#
#
test "$DEBUG" = True && set -x
export system_password=coolrunnings
export sys_password=helloworld
export scott_passwd=tiger
export app_user=psoft
export oracle_sid=FS75PROD
export oracle_owner=oracle
```

erase set the erase terminal character. Normally the system can determine the erase character by itself. However, in some circumstances it is still wrong. The **erase** command allows you to choose an erase character from a list or to specify the correct character by yourself.

header create, edit, view, remove a menu header. The menu header is intended to display additional information below the menu title, if the menu title information is not sufficient.

help, h, usage, ?
print a short help of all internal and contributed (**contrib.edrc(1m)**) commands.

history, hist
print the history of **edrc**. The history contains all invoked commands. Commands which are entered repetitive are listed only once.

hostname print the hostname of the system you started **edrc** on.

loadrec load a previous saved recovery point in time from disk. This will set the **\$EDRC_RECOVERYTIME** environment variable which is exported to all recovery scripts. To display a recovery point in time, use the **printrec** command, to set a recovery point in time, use the **setrec** command.

log view the last logfile. Every output of a recovery script is saved to a logfile. Therefore your recovery scripts don't have to care about logging information.

man display manual pages.

menu, m, ls

print the menu. In the menu title the name of the current session (if set with the **name** command) is displayed. All menupoints are enclosed by square brackets: [28]. Submenus are marked with a " > " sign. The current location in the script tree is part of the **edrc** prompt. The prompt consists of three parts which might not be present at all times:

edrc@host/Oracle/PSYS:157>

In this example you are in the sub-sub menu "PSYS" and the last menupoint you selected was "157". If you execute a **edrc** command after you selected a menupoint this will not influence the prompt. So it is very easy to see what your last action during the recovery was.

It will make sense that the menupoints correspond to the reference numbers in the "Emergency Handbook" of your environment.

name give the current session a name. This is useful to avoid confusion and to find the right session if you started parallel multiple sessions.

The logfiles contain the session name in square brackets: [example].

If you do not set a session name the current process id or a random value is used as an (ad hoc) session name.

The current session name is printed in the menu head (SESSION:) when the menu is listed using the **menu** command.

During a "Disaster Case" or "Disaster Case Simulation" it is highly recommended to set the session name as advised by the "Emergency Handbook".

If possible a session name is transferred to a remote host when a trunk is established. However, if on the remote system a session with the same name is already running, a new (ad hoc) session name will be set automatically.

newmenu, newdir

create a new menu in the recovery script menu tree. **edrc** will guide you thru the naming of the menu. Finally a menu is a directory with the name:

<menu_point>:<name>

menu_point *and* *name* has to be unique in the current menu.

If the menu title contains a semicolon (;) the menu entry will be broken into two lines at the position of the semicolon when using the **menu** command.

newscript create a new recovery script. **edrc** will guide you thru the naming of the script and will provide a template script which contains some key elements. The name of the script will be:

<menu_point>:<name>

menu_point *and* *name* has to be unique in the current menu.

If the script header contains a semicolon (;) the menu entry will be broken into two lines at the position of the semicolon when using the **menu** command.

PID print the UNIX process id of the **edrc** session.

printrec display the currently set recovery point in time. To set a recovery point in time, use the **setrec** command, to load a saved recovery point in time, use the **loadrec** command.

quit quit **edrc** at any point.

removemenu
remove an existing submenu. Prior to the removal the menu is backuped.

removescript
remove an existing recovery script. Prior to the removal the file is backuped.

renamemenu
rename an existing submenu.

renamescript
rename an existing recovery script. Prior to the rename the file is backuped.

setrec set recovery point in time. This will set the **\$EDRC_RECOVERYTIME** environment variable which is exported to all recovery scripts.

The date format has to be compliant to the **NLS_DATE_FORMAT** configured in **edrc.cfg**.

You can save your recovery point in time to have it available after a newstart of **edrc**. To save the recovery point in time would be a good idea in most cases. If you have to change the recovery point in time only temporarily for a certain recovery action you probably won't save it.

To display a recovery point in time, use the **printrec** command, to load a saved recovery point in time, use the **loadrec** command.

shell start a shell in **edrc**. You should use this command to exit to the operating system rather than to do an other telnet or rlogin to the system.

When you use the **shell** command the current working directory is the location in the menu you are currently in. Furthermore the shell settings are tuned and customized to the WA2L/edrc environment and all **\$EDRC_** environment variables are exported to the shell.

On UNIX the Korn-Shell (ksh), on Linux the Bourne-Again-Shell (bash) will be started. To see all additional commands and aliases you have available in the shell type **usage** within the started shell.

See **shell(1)** for a description of the additional commands available.

term set the terminal emulation (the **\$TERM** environment variable) for the current session. See also: **edrc.cfg(4)**.

title edit, view, remove a menu title created while creating a new submenu. Or create a menutitle for a menu which was not labeled so far.

top browse directly back to the root menu. If it is a trunk, browse to the trunk root.

trunk create, edit, view, remove, activate, deactivate the trunk definition (**_trunk**) for a menu. A trunk is basically the invocation of a menu on a remote system. Therefore, ahead of defining a trunk, a normal menu has to be created. A trunk definition enhances a menu with the behavior to be started on a remote system transparently without the need to first manually log on to the remote system, start **edrc** and then change to the desired submenu.

To establish a trunk to a remote system it must be possible to connect with *OpenSSH* mode (**-m OpenSSH**) to the remote system using the **remote_shell** command and the recovery scripts have to be distributed. See also **remote_shell(3)** and **edrc.cfg(4)** for more information.

When a new trunk is defined using the **trunk** command, a specification template consisting of the following elements is provided:

target::= [*comment* :] *hostname*

The *target* listed in the specifications below can consist of a *comment* (optional) and a *hostname* (not optional) where to connect to. The *comment*, if specified, is displayed at the trunk dialog instead of the *hostname*. The *comment* and the *hostname* has to be separated by a colon (:). Furthermore spaces and tabs are not allowed in a *comment* text and between the *comment*, the : and the *hostname*. If spaces in the *comment* output are needed, they have to be replaced by a % character in the definition.

ACTIVE= *True* | *False*

If **ACTIVE** is set to *True* the trunk specification is active and a remote menu invocation will be established based on the remaining trunk settings. If it is set to *False* the trunk specification exists but is not active. In this case the menu behaves as a normal submenu. When the **trunk** command is invoked, it is possible to toggle this setting using the (a)ctivate and (d)eactivate options prompted when invoking the **trunk** command.

TYPE= *direct* | *prompt* | *prompt:nolocal*

If **TYPE** is set to *direct* the trunk is established immediately without prompting the user. When it is set to *prompt* or *prompt:nolocal* a dialog is displayed and the user is prompted for a selection of a target to establish the trunk to. The dialog for a *prompt* trunk always includes one menu point to stay on the local system and not to establish a trunk to a remote system. A *prompt:nolocal* trunk does not provide the local selection.

DEFAULT= " *target* "

The target definition here takes into effect if the **TYPE** setting is set to *direct* and no *target* is entered in the **TARGETS** definition. If **TYPE** is set to *prompt* the setting made here is printed in the "Default:" section of the user dialog. This setting is interpreted by a shell, therefore it is also possible to do some dynamic stuff here.

TARGETS= " *target* [{ *target* }]"

White space separated list of targets to be listed in the "Targets:" section of the user dialog when **TYPE** is set to *prompt*. If **TYPE** is set to *direct* and no **DEFAULT** target is defined, all elements of this list are probed and the first one that is up is used to establish the trunk to. This setting is interpreted by a shell, therefore it is also possible to do some dynamic stuff here.

TARGET_MENUPATH= " *menupath* "

This is an optional setting that is only used for advanced exceptional trunk specifications. When specifying the *menupath* it is possible to choose a different entry menu at the trunk target.

TARGET_CONFIG= " *configfile* "

This is an optional setting that is only used for advanced exceptional trunk specifications. When specifying the *configfile* it is possible to establish a trunk to a target using a target configuration that differs from the local configuration.

1) Example of a simple trunk specification:

```
#
# _trunk - Trunk specification for menu /HP_11/services
#
# [00] 22.03.2007 CWa Initial Version
#
#
ACTIVE=True
TYPE=prompt
DEFAULT="host-001"
TARGETS="host-001 host-002 host-003 host-004"
```

2) Example of a more enhanced trunk specification:

```
#
# _trunk - Trunk specification for menu /HP_11/services
#
# [00] 22.03.2007 CWa Initial Version
#
#
```

```

ACTIVE=True
TYPE=prompt
DEFAULT="'server_environment':localhost"
TARGETS="
    MAINTENANCE:host-001
    TEST:host-002
    PREPRODUCTION:host-003
    PRODUCTION:host-004
"

```

3) Example defining a dynamic target list

The target list consists of all hosts defined to be member of the group `@ALL` as defined in the `HOSTGRPS` setting in the `edrc/etc/hostlist.cfg` file. Therefore, if a host is added to the `@ALL` group, the trunk target list will be automatically enhanced.

```

#
# _trunk - Trunk specification for menu /HP_11/services
#
# [00] 22.03.2007 CWa Initial Version
#
#
ACTIVE=True
TYPE=prompt
DEFAULT=""
TARGETS="'hostlist -g @ALL'"

```

4) Example of exceptional advanced trunk specification

Suppose `~edrc/bin/sat` is started (`~edrc/sbin/edrc -c edrc.sat.cfg`), you are in the menu `/HP_11` and the submenu `services` is a trunk. This example definition will establish a trunk using the configuration `~edrc/etc/edrc.psup.cfg` on the trunk target. The entry menu on the remote system is `/user_mgmt` instead of `/HP_11/services`. It is not imperative to use both advanced trunk specification options in an advanced trunk definition.

Hint: The advanced trunk definition options (`TARGET_MENUPATH` and `TARGET_CONFIG`) should be used **very** moderately and with a good reason, due to the fact that the chance is high that advanced trunks cannot be established later on (error message: **edrc-FATAL: menu does not exist, aborting**), because the target menu structure could be changed without realizing that a certain (sub)menu is a trunk target. On normal trunks, where the menu and the configuration is identical and the menu tree is distributed, this cannot happen.

```

#
# _trunk - Trunk specification for menu /HP_11/services
#
# [00] 22.03.2007 CWa Initial Version
#
#
ACTIVE=True
TYPE=prompt
DEFAULT="host-001"
TARGETS="host-002 host-003 host-004"
TARGET_MENUPATH="/user_mgmt"

```

```
TARGET_CONFIG="edrc.psup.cfg"
```

trunkbypass

toggle bypassing of active trunks between *True* and *False*. When toggle trunk bypassing to *True* all active trunks are bypassed in the current session and are not followed.

It might be helpful to bypass trunks to efficiently edit many menu points in submenus which are defined as trunks. Doing so you do not have to first deactivate and after editing activate the trunk for each menu separately.

tty print the TTY you are logged on and the **edrc** program is started in.

tz set the time zone (the **\$TZ** environment variable) for the current session. See also: **edrc.cfg**(4) and **timezone**(3).

ENVIRONMENT

The following environment variables are exported to all recovery scripts, contributed commands and the shell. Do not use other variables seen by your scripts which are related to the WA2L/edrc environment because they may disappear in future releases or its content may change without notice.

Use the **edrcenv** command within the **shell** or **edrc** to see the current environment settings.

\$EDRC_NLS_DATE_FORMAT

national language support date format. The recovery point in time has to comply with this format.

\$NLS_DATE_FORMAT

same as **\$EDRC_NLS_DATE_FORMAT**.

\$EDRC_NLS_LANG

national language support language name.

\$NLS_LANG

same as **\$EDRC_NLS_LANG**.

\$DEBUG

used to debug recovery scripts or contributed commands. This variable is set to *False* (no debugging) or *True* (debugging). Your script should contain the following line to allow debugging:

```
test "$DEBUG" = True && set -x
```

\$EDRC_ENTRY_DIR

current directory where your recovery script is located or where you enter the filesystem tree when you invoke the **shell** command.

\$EDRC_TMP_DIR

this variable points to a secure temporary directory that exists during the execution of a recovery script, contributed command or the shell command and is afterwards removed. This variable is therefore exported to recovery scripts, contributed commands and the shell command only.

This variable should be used to securely save temporary data during the execution of a recovery script or a contributed command.

\$EDRC_TMP_DIR_SESSION

this variable points to a secure temporary directory that exists during the execution of an **edrc** session and is afterwards removed. This variable is exported to recovery scripts, contributed commands and the shell command only.

This variable should be used to securely save temporary data from recovery scripts or contributed commands that must be accessed multiple times throughout the **edrc** session.

You should add the name of the recovery script or contributed command as prefix to ensure a unique filename.

Example:

```
date > $EDRC_TMP_DIR_SESSION/$EDRC_SCRIPTNAME.begin
```

If you need to save temporary data during the execution of a recovery script or a contributed command the **\$EDRC_TMP_DIR** variable shall be used.

\$EDRC_OSID

the systems operating system id evaluated by **osid**. See **osid(3)** for more information.

\$EDRC_SCRIPTS_BASEDIR

name of the recovery scripts basedir. This environment variable is set to the value you specified with the **SCRIPTS_BASEDIR** option in the **edrc.cfg** configuration file. See **edrc.cfg(4)** for more information.

\$EDRC_ENV

name of the environment file. See description in section **COMMANDS**, at the **env** command. Therefore do **not** hard-code **_env** in your scripts, use this variable.

\$EDRC_CONFIGFILE

name of the configuration file loaded in the current **edrc** session.

\$EDRC_LOGFILE

logfile of the current script. Normally you will not have to write to this file by your own, **edrc** will redirect all output of your recovery script to this file.

\$EDRC_RECOVERYTIME

point in time to recover to. See description in section **COMMANDS**, at the **setrec** command.

\$EDRC_SCRIPTNAME

name of the recovery script (without the "<menu_point>:" part). If you use the **msg** command for message output this name will appear in front of the "-" sign.

\$EDRC_MENUPPOINT

menu point of the recovery script (without the ":<scriptname>:" part).

\$EDRC_SESSION

name of the **edrc** session. If you use the **log** command (what will normally not be the case in recovery scripts) the session name is written between the square brackets into the logfile (**\$EDRC_LOGFILE**).

\$EDRC_CONTRIB_VARDIR

var directory for contributed commands. This environment variable is only exported to contributed commands. If you have to save persistent information store it to this location.

You should add the name of the contributed command as prefix to ensure a unique filename.

Example:

```
date > $EDRC_CONTRIB_VARDIR/$EDRC_SCRIPTNAME.begin
```

\$EDRC_SEED

session persistent random seed number.

\$EDRC_SHELLHIST

history file for the shell started with the **shell** EDRC command. This environment variable is only exported to the shell started within **edrc**.

\$EDRC_DIST_USER

this variable equals to the **DIST_USER** setting in the **etc/edrc.cfg** file.

\$EDRC_DEBUG

switch all commands in **edrc/bin** and **edrc/lib** into debugging mode. This allows to debug also commands that don't have the debugging option **-x** implemented.

Do not mix up this environment variable with the **\$DEBUG** environment variable. When debugging recovery scripts only the **\$DEBUG** variable is of significance.

\$EDRC_DEBUG_COMMAND

switch a *command* in **edrc/bin** or **edrc/lib** into debugging mode. This allows to debug also commands that don't have the debugging option **-x** implemented.

Do not mix up this environment variable with the **\$DEBUG** environment variable. When debugging recovery scripts only the **\$DEBUG** variable is of significance.

\$TODAY current date in the form YYYYMMDD (example: 20061021).

SIGNALS

The following signals are handled by **edrc**. Do not use other signals as those listed below, as long as you do not really know what you are doing and what the consequences are.

- INT** interrupt from keyboard (Ctrl+C).
If **ABORT_MODE** is set to *hard* **edrc** will abort completely, if **ABORT_MODE** is set to *soft* only the current running command or recovery script will be aborted.
- TERM** kill the process without an argument.
If **ABORT_MODE** is set to *hard* **edrc** will abort completely, if **ABORT_MODE** is set to *soft* only the current running command or recovery script will be aborted.
- USR1** the **edrc** command will perform a "soft" abort, as if **ABORT_MODE** is set to *soft*.
- USR2** the **edrc** command will perform a "hard" abort, as if **ABORT_MODE** is set to *hard*.

EXIT STATUS

- 0** no error.
- 1** you started **edrc** using the wrong operating system user.
- 2** operating system is not supported. See **osid(3)** if you get this error.
- 3** version printed.
- 4** usage listed.
- 5** program aborted (Ctrl+C ...).
- 6** the configuration file does not exist.
- 7** the directory specified in **SCRIPTS_BASEDIR** does not exist.
- 8** **edrc** was started using the **-n session_name** option and another session with the same *session_name* is already running on this host.

- 9 the **edrcrevision.cfg** file is missing or the **VERSION** or **PATCHLEVEL** options within this file are not set properly.
- 10 **edrc** was started in trunk mode from a remote system and the trunk root menu does not exist.
- 11 a temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

The **name(1)** command can be used to list a short description of files and directories.

etc/edrc.cfg

configuration file of **edrc**, see **edrc.cfg(4)** for more information.

etc/edrcrevision.cfg

configuration file which holds the current version and patchlevel of EDRC. See **edrcrevision.cfg(4)** for more information.

etc/exrc

configuration file for the vi-Editor and viewer started within **edrc**. See **svi(1)**, **exrc(4)** and **vi(1)** for additional information.

etc/shell.cfg

generic configuration file for the shell started by the **shell** command. This file is read by the **etc/kshrc** shell startup file. The intention of this configuration file is to be able to configure some shell variables without changing the more complicated **kshrc** file.

etc/kshrc

startup file for the shell started by the **shell** command. Normally you won't have to change this file, changing configurations are handled in the **etc/shell.cfg** file. See **kshrc(4)** for more information.

contrib/edrc/

contributed commands to edrc. Do not change the **edrc** program if you miss a general command. Create the command in this directory and you will be able to invoke it as it is an **edrc** internal command. This will protect you from losing your work after an upgrade of **edrc** and will keep the **edrc** command itself stable.

lib/

commands which are mostly used within scripts. Some of them require some environment settings. This directory is in \$PATH

lib/edrc/

library functions called by the **edrc** command.

scripts/

here are all recovery scripts stored by default. The content of this directory has to be customized to your environment. For a description how to write scripts for EDRC see **edrc-scripts(1M)** or the example configuration included in the WA2L/edrc package.

.sav

backup directory in a recovery script tree. Whenever a file in **edrc** is edited or removed a backup copy is automatically saved into the **.sav** directory of the current menu (=subdir). Therefore, if you like to restore a file to a prior version, invoke the **shell** command in **edrc**,

change to the **.sav** directory and restore the desired file using the normal operating system commands. The **.sav** directory is excluded from distributions when invoking the **distribute** command.

var/backup/

several backups.

var/contrib/edrc/

var directory for contributed commands. The **\$EDRC_CONTRIB_VARDIR** environment variable points to this directory. See section **ENVIRONMENT** for information how to save persistent information in contributed commands.

var/lock/

lockfiles. Do not edit them by hand.

var/log/

all logfiles of the EDRC environment. Here the logfile of **edrc** and the output of all recovery scripts is saved. Therefore customizing scripts don't have to implement an own logging mechanism. All output sent to **stdout** and **stderr** will be saved in a designated logfile.

var/repl/

this directory is used to save scripts replicated with the **distribute** EDRC command.

var/connection/security/<DIST_USER>/<DIST_MODE>/<host_from>|default/<host_to>|default}

this directory holds security information used for example by secure shell connections. See **edrc.cfg(4)** for more information about the currently possible **DIST_MODE** settings and **remote_shell(3)** and **remote_copy(3)** for more information about the **var/connection** directory.

var/connection/cache/<DIST_USER>

this directory holds the cache of successful connections. This is used to speed up the initialization of a connection due to the fact that the first try is to use the **DIST_MODE** first which resulted already in a successful connection.

var/repl/own.<cfg>.full.cpio.gz

full distribution file created on the local host. This archive contains all scripts from **SCRIPTS_BASEDIR** defined in the configuration file *cfg*.

var/repl/rec.<cfg>.full.cpio.gz

full distribution file created on a remote host and distributed to the local host. This archive contains all scripts from **SCRIPTS_BASEDIR** defined in the configuration file *cfg*.

var/repl/own.<cfg>.diff.cpio.gz

differential distribution file created on the local host. This archive contains all scripts from **SCRIPTS_BASEDIR** defined in the configuration file *cfg* which are newer then the most recent full distribution file available on the local host.

var/repl/rec.<cfg>.diff.cpio.gz

differential distribution file created on a remote host and distributed to the local host. This archive contains all scripts from **SCRIPTS_BASEDIR** defined in the configuration file *cfg* which are newer then the most recent full distribution file available on the remote host where the distribution file comes from.

var/settings/

persistent settings.

var/shell/ Here the shell history of each session of the interactive shell built in to **edrc** and the own history of **edrc** is kept.

SEE ALSO

edrcintro(1), **edrc.cfg(4)**, **edrcrevision.cfg(4)**, **bash(1)**, **catcomp(1)**, **contrib.edrc(1m)**, **Emergency Handbook**, **envpasswdstrip(1)**, **hostlist(3)**, **hostlist.cfg(4)**, **ksh(1)**, **llcomp(1)**, **lscomp(1)**, **lscp(1)**, **lsmv(1)**, **name(1)**, **osid(3)**, **remote_copy(3)**, **remote_shell(3)**, **scriptgrep(1)**, **sh(1)**, **shell(1)**, **vi(1)**, **view(1)**

NOTES

-

BUGS

-

AUTHOR

edrc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2021 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrc.cfg – configuration file for edrc

SYNOPSIS

edrc/etc/edrc.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **edrc** command.

It is important that on all hosts in your disaster recovery environment the configuration files with the same purpose have the same name. Otherwise the **distribute** EDRC command won't function correctly.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SCRIPTS_BASEDIR**

name of the recovery scripts basedir. In this directory all recovery scripts for your environment are stored in a hierarchical directory tree. This directory is relative to the root of the WA2L/edrc installation.

Example: **SCRIPTS_BASEDIR=scripts/customer**

Default: **SCRIPTS_BASEDIR=scripts**

SCRIPTS_SHELL

path to the shell you will use the most in your recovery scripts located in the **SCRIPTS_BASEDIR**. This setting will be used to build a template script when you invoke the newscript command in **edrc**.

Example: **SCRIPTS_SHELL=/bin/ksh**

Default: **SCRIPTS_SHELL=/bin/sh**

SCRIPTS_SHELLFALLBACKS

fallback shell definitions for magic keys of recovery scripts. This fall-backs come into action, if the shell defined in the magic key of a recovery script does not exist on the system. Be

aware, that only for magic keys without options the fallback shell will be resolved. For scripts having options the script must be edited and the correct shell has to be defined in the script. The entry is a list in the format `<original_shell>:<fallback_shell>`. The fallback resolution is performed in the order of the entries. Before resolving the entries in the **SCRIPTS_SHELL_FALLBACKS** list, the shell as defined in the recovery script is searched in alternate locations on the system. For all shell location resolutions the **\$PATH** is used.

Example: `SCRIPTS_SHELLFALLBACKS="ksh:pfksh ksh:bash sh:ksh sh:pfsh sh:bash"`

Default: `SCRIPTS_SHELLFALLBACKS="ksh:bash sh:ksh sh:bash"`

NLS_DATE_FORMAT

NLS date format. This format is exported to each recovery script. You should set your most used format here. If you have exceptions to this format you can set it for each EDRC menu with the **env** EDRC command.

Example: `NLS_DATE_FORMAT='MON DD YYYY HH24:MI:SS'`

Default: `NLS_DATE_FORMAT='MON DD YYYY HH24:MI:SS'`

NLS_LANG

NLS language name. This name is exported to each recovery script. You should set your most used format here. If you have exceptions to this format you can set it for each EDRC menu with the **env** EDRC command.

Example: `NLS_LANG=american`

Default: `NLS_LANG=american`

TERM

Terminal emulation used. You should set this setting to the terminal you use the most. You can change the **TERM** setting using the **term** EDRC command for an **edrc** session.

Example: `TERM=dtterm`

Default: `TERM=vt220`

LOG

Log output dir of **edrc**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: `LOG=/var/opt/edrc/log`

Default: `LOG=var/log`

LOGCOMPRESS

Compress logfiles of recovery scripts and contributed commands. If this setting is set to *False* the logfiles will not be compressed, if it is set to *True* the logfiles will be compressed after completion of the recovery script or contributed command. Using this setting you can save a significant amount of disk space. To view the logfile you can still use the **log** command or the **logs** contributed command within **edrc** without having to care if a log file is compressed or not. If you go to the **edrc/var/log** directory you can use **zgrep**(1) to grep the compressed logfiles.

Example: `LOGCOMPRESS=True`

Default: LOGCOMPRESS=False

LOGHEAD_OUTPUT

Output of the log header. If this setting is set to *False* the log header will only go to the logfile, if it is set to *True* it will go to the screen and to the logfile.

Example: LOGHEAD_OUTPUT=False

Default: LOGHEAD_OUTPUT=True

START_USER

This is the user **edrc** has to be started with.

Example: START_USER=root

Default: START_USER=root

DIST_HOSTLIST

Space separated list of hosts. To this hosts the files in **SCRIPTS_BASEDIR** will be distributed.

Example: DIST_HOSTLIST="acme045 acme014"

Default: –

DIST_USER

User used to distribute the files in **SCRIPTS_BASEDIR**. The home of this user is considered as the root of the **WA2L/edrc** installation. See **edrcsetup**(1m) for information about user settings needed by **WA2L/edrc**.

Example: DIST_USER=edrc

Default: DIST_USER=edrc

EDRC_OWNER

Owner of the **WA2L/edrc** software. This is the user the EDRC environment is installed with. See **edrcsetup**(1m) for information about user settings needed by **WA2L/edrc**.

Example: EDRC_OWNER=root

Default: EDRC_OWNER=root

DENY_LIST

Space separated list of functions/commands to be denied from invocation in EDRC. Be aware, that certain denials made here could be by-passed dependent of the operating system by the built in functionality of the **more**(1) and **vi**(1) commands used by EDRC internally. Be also aware, that potential sources of by-passing the denials are also the contributed commands, and of course the recovery scripts.

To minimize possible by-passes in contributed commands and in recovery script configurations (used in shared configurations with partly restricted rights) the **is_permitted**(3) command should be used to control command startup.

Currently the following denials are available: *banner, browse, busy, clear, contrib, contrib.*, date, debug, distribute, edit_file, edrcdebug, edrcperm, env, env.edit, env.more, env.new, env.remove, env.view, erase, header, header.edit, header.more, header.new, header.remove, help, history, hostname, loadrec, log, log.grep, log.head, log.more, log.tail, log.view, man, menu, more_file, name, new_file, newmenu, newscript, PID, printrec, recoveryscript.*, remove_file, removemenu, removescript, renamemenu, renamescript, script, script.edit, script.more, script.start, script.view, setrec, shell, term, title, title.edit, title.more, title.new, title.remove, trunk, trunk.change, trunk.edit, trunk.follow, trunk.more, trunk.new, trunk.remove, trunkbypass, tty, tz, view_file.*

Denials represented by a single word deny access to a command as a whole. If a denial consists of two words separated by a point, this will deny access to a sub function of a command. Denials containing an underscore deny access to low level functions that are used by many commands or command functions. Therefore if you like to deny any file editing in **edrc** you should add *edit_file* to the **DENY_LIST** instead of listing all *<command>.edit* denials.

Honestly, I could not imagine why to use many of those denials, but it gives you the maximal flexibility of configuration.

Example: `DENY_LIST="newscript newmenu contrib.omnimon"`

Default: `DENY_LIST=""`

DIST_MODE

Comma separated list of modes used to distribute the files in **SCRIPTS_BASEDIR**. The supported modes are: *rtools* which result in the use of **rcp** and **rsh** or *OpenSSH* which results in the use of **scp** and **ssh** for distribution. If a comma separated list is provided, the connection initiation is made in the sequence specified. A pseudo distribution mode is *default* which results in the use of the **CONNECTION_MODE** specified in the configuration files **remote_shell.cfg** and **remote_copy.cfg**. It is not allowed to specify *default* as part of a comma separated list.

Example: `DIST_MODE=rtools,OpenSSH`

Default: `DIST_MODE=rtools`

DIST_TYPE

Type of distribution. The supported types are: *serial* which results in a host by host distribution or *parallel:max* which results in a parallel distribution to multiple hosts. The *max* option specifies the maximal number of parallel distribution sessions.

Example: `DIST_TYPE=parallel:20`

Default: `DIST_TYPE=serial`

EDRC_OWNER

Owner of the WA2L/edrc software. This is the user the EDRC environment is installed with. See **edrcsetup(1m)** for information about user settings needed by WA2L/edrc.

Example: `EDRC_OWNER=root`

Default: `EDRC_OWNER=root`

EDRC_PROMPT

Prompt to be displayed before the @ symbol in **edrc**.

Example: `EDRC_PROMPT=sysadmin`

Default: `EDRC_PROMPT=edrc`

EDRC_HOSTPROMPT

With this setting it can be chosen, if the host name in the prompt that is displayed after the @ symbol in **edrc** is displayed in full length (*long*), or if only the host name part (*short*) is displayed.

Example: `EDRC_HOSTPROMPT=short`

Default: `EDRC_HOSTPROMPT=long`

BACKUP_DIR

This setting is used by **scriptextract** only. **scriptextract** saves all script files ahead of extracting (installing) the newly distributed scripts to this directory.

Example: `BACKUP_DIR=/var/opt/edrc/backup`

Default: `BACKUP_DIR=var/backup`

BACKUP_GENERATIONS

This setting is used by **scriptextract** only. Here the number of kept script backup versions can be specified. If this setting is set to 0, all versions will be kept and no backup purging will take place.

Based on experiences with systems having WA2L/edrc installed for about five years, the space needed to keep all script backup versions is not huge in average. Therefore this value can be set to a reasonable high value or to 0 without having bad feelings.

Example: `BACKUP_GENERATIONS=200`

Default: `BACKUP_GENERATIONS=0`

TMP_DIR

Specify the location of temporary files generated by **edrc** and **scriptextract**. It is only needed to change this setting from the default value if you experience filesystem fill ups on the default location. An alternate temporary directory has to have the identical permissions as the default location */tmp*.

Example: TMP_DIR=/var/tmp

Default: TMP_DIR=/tmp

CONTRIB_MESSAGE

Normally ahead of executing a contributed command the message "edrc-INFO: starting contributed command 'example'" is printed. To switch this message off, change this setting from *True* to *False*.

Example: CONTRIB_MESSAGE=False

Default: CONTRIB_MESSAGE=True

NOTRECOGNIZED_MESSAGE

If this option is set to *True* the message "edrc-WARNING: 'example' is not recognized as a command or menu point." is printed if the user enters a command which is neither an internal/contributed command nor a menu point in the current recovery script menu. To switch this message off, change this setting from *True* to *False*.

Example: NOTRECOGNIZED_MESSAGE=True

Default: NOTRECOGNIZED_MESSAGE=False

RECOVERYPOINTINTIME_MESSAGE

If this option is set to *True* the message "edrc-WARNING: recovery point in time is *NOT* set!" is printed if the user starts a command and the recovery point in time is not set. It will not make sense to set this option to *False* in a disaster recovery script tree. However, if you use **sat** or an other configuration whose main purpose is not disaster recovery but system administration where you'll never use the "recovery point in time" it might be useful to not to print this message. To switch this message off, change this setting from *True* to *False*.

Example: RECOVERYPOINTINTIME_MESSAGE=False

Default: RECOVERYPOINTINTIME_MESSAGE=True

SETENVIRONMENT_MESSAGE

If this option is set to *True* the message "edrc-INFO: ahead of starting 'a:example' the environment defined in '_env' will be set" is printed if the user starts a command and an environment file exists for the current menu. It will not make sense to set this option to *False* in a disaster recovery script tree. However, if you use **sat** or an other configuration whose main purpose is not disaster recovery but system administration where you don't want to show ahead of each command start the message if an environment will be set it might be useful to not to print this message. To switch this message off, change this setting from *True* to *False*.

Example: SETENVIRONMENT_MESSAGE=False

Default: SETENVIRONMENT_MESSAGE=True

ABORT_MODE

Abort mode. This setting influences the abort behavior of edrc. If edrc receives the INT (Ctrl+C) or TERM (kill) signal the program will abort completely if **ABORT_MODE** = *hard*, **edrc** will only abort the current running script or command if **ABORT_MODE** = *soft*.

Example: ABORT_MODE=soft

Default: ABORT_MODE=hard

HISTORY_LENGTH

edrc saves every command you enter in a history. When you issue the **history** EDRC command a list of your last invoked commands is displayed. The length of this list can be configured with this option.

Example: HISTORY_LENGTH=20

Default: HISTORY_LENGTH=100

HISTORY_SEARCH

Number of history lines to be searched for a numbered command. The history is searched from the end to the beginning.

Example: HISTORY_SEARCH=200

Default: HISTORY_SEARCH=100

HISTORY_BROWSE

Control if the history can be accessed using the cursor keys and **vi** controls (**j**, **k**, **/**).

On systems with **OSID=*Linux*** the **HISTORY_BROWSE** is turned on (**=True**) by default (if the system supports it), on other OSIDs it's turned off.

Example: HISTORY_BROWSE=False

Default: HISTORY_BROWSE=True (for OSID *Linux*)

Default: HISTORY_BROWSE=False (for other OSID)

MAN_PATH

Additional man pages which are not available in the \$MANPATH of the user defined in **START_USER**. This additional manpages will be appended to the current \$MANPATH and are accessible thru the **man** command within EDRC. Do not add the man pages of EDRC to this path, those pages are available by default.

Note: This setting has become of minor importance. If all defined manual pages should be displayed in all **edrc** configurations identically, and also in **man** (**edrcman**), the manual page locations should be listed in the **edrcman.cfg** file.

Example: MAN_PATH=/opt/omni/man:/opt/apl/asystools/man

Default: MAN_PATH=-

SEE ALSO

edrc(1m), **edrcintro(1)**, **is_permitted(3)**, **maketemp(3)**

NOTES

-

BUGS

-

AUTHOR

edrc.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrc.help – help file for edrc

SYNOPSIS

edrc/lib/edrc/edrc.help

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the help file for the **edrc** command. If you invoke the **help** EDRC command, this plain ASCII file will be displayed.

FILEFORMAT

Plain ASCII file. The columns are aligned using tabs with the width of 4.

SEE ALSO

edrcintro(1), **edrc(1m)**

NOTES

-

BUGS

-

AUTHOR

edrc.help was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcinit – WA2L/edrc services startup/shutdown/restart/status

SYNOPSIS

edrc/bin/edrcinit [**-h**]

edrcinit (**start** | **stop** | **restart** | **refresh** | **status** | **statuslist**) [*name*]

edrcinit list | **guard** | **unguard** | **checknow** | **rcinstall** | **enable** | **disable**

edrcinit (**cfg-activate** | **cfg-unactivate** | **cfg-guard** | **cfg-unguard**) *name*

AVAILABILITY

WA2L/edrc

DESCRIPTION

handle services start/stop/restart, primarily for daemons included in WA2L/edrc, but other services can be handled as well. **edrcinit** supports to monitor and to restart a failed service based on the definitions made in the **edrcinit.cfg** config file.

OPTIONS

start start WA2L/edrc service(s).

stop stop WA2L/edrc service(s).

When a guarded service is stopped using '**edrcinit stop name**' it is not guarded until '**edrcinit start [name]**' is invoked.

restart restart WA2L/edrc service(s). A **restart** is equal to a '**edrcinit stop; edrcinit start**' invocation.

refresh restart a non interruptive service restart. In the **edrcinit_handler** a restart should be implemented that way, that the targeted service receives a command or signal, that causes a non-interruptive restart. This is often implemented by reacting to a received **SIGHUP** signal.

status print status of all activated service(s).

statuslist print status of all service(s) in list form.

name service name. If this option is not specified, all defined and activated service(s) are addressed.

- list** list all defined service(s).
- guard** guard the service(s). With this option it is possible to switch on the guarding of the services, that are defined to be guarded. This option is only used after switching off the service guarding using the '**edrcinit unguard**' command.
- unguard** Switch the guarding of the services globally off. After a newstart of **edrcinit** or the invocation of '**edrcinit guard**' the guarding is switched on again.
- checknow** force an immediate service check.
- rcinstall** install a small rc script to the system related **init** directory (**/etc/init.d/**, **/sbin/init.d/** or **/etc/init/**) and ensure that **edrcinit** is started on system boot.
- enable** globally enable the start of services handled thru **edrcinit** after the startup has been disabled using **edrcinit disable**.
- The prerequisite to ensure the start of **edrcinit** is the installation of the related rc script thru the invocation of **edrcinit rcinstall**.
- disable** globally disable the start of **edrcinit** even if **edrcinit rcinstall** has been invoked, but without the need to disable the service startup system wide.
- Even when the service startup has been disabled using the **edrcinit disable** command, services can still interactively be started, stopped, etc. using the **edrcinit { start | stop | ... }**
- cfg-activate** *name*
set *activated* in the configuration file for the service *name* to **Y**.
- cfg-unactivate** *name*
set *activated* in the configuration file for the service *name* to **N**.
- cfg-guard** *name*
set *guarded* in the configuration file for the service *name* to **Y**.
- cfg-unguard** *name*
set *guarded* in the configuration file for the service *name* to **N**.

ENVIRONMENT

\$PRINT_FIT2WIDTH

If not set to *False*, the output when invoking the **list** option is limited to the current width of the terminal window and rows extending the window width are marked with '>>'. If set to *False* the whole output line is printed and wrapped at the line end if applicable.

EXIT STATUS

- 0** no error.
- 1** error in start/stop/restart/refresh/status of specified service(s) or guard/unguard of services(s).
- 2** operating system not supported.
- 3** configfile does not exist.
- 4** usage listed.
- 5** specified service name (*name*) is not defined or is not activated.
- 8** **edrcinit** cannot write to logfile.
- 11** temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

- etc/edrcinit.cfg**
config file of **edrcinit**.
- lib/edrcinit/**
directory for handlers of the service(s) handled thru **edrcinit**. See **edrcinit.handler(3)** for additional information.
- lib/edrcinit/WA2Ledrc.edrcinit.check**
check script of watchdog to guard the services in **edrcinit**.
- lib/edrcinit/WA2Ledrc.edrcinit.bite**
bite script of watchdog to guard the services in **edrcinit**.
- var/log/edrcinit.log**
log file of **edrcinit**.
- var/edrcinit/settings**
persistent data of **edrcinit**.
- var/edrcinit/stopped**
list of stopped services using the '**edrcinit stop** [*name*]' command.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcinit.cfg(4)**, **edrcinit.handler(3)**, **kill(1)**, **signal(7)**, **vsav(1)**, **watchdog(1)**

NOTES

The guarding of the services is handled thru a watchdog with the name **EDRCINIT** which is started thru **edrcinit**. See **watchdog(1)** for more information.

BUGS

edrcinit is tested on Solaris, HPUX, RedHat Linux, SuSE Linux, Fedora Linux and Ubuntu Linux and supports the specific startup methods of the related operating system.

AUTHOR

edrcinit was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcinit.cfg – configuration file for edrcinit

SYNOPSIS

edrc/etc/edrcinit.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **edrcinit** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The fileformat for the '**GENERAL OPTIONS**' is **OPTION=VALUE**

Between the **OPTION**, the **=** and the **VALUE** are no spaces.

You should not comment out any **OPTION**. If you like to use default settings simply do not specify a **VALUE**.

For the '**SERVICE HANDLING OPTIONS**', the fileformat is *field ; field ; field ; field ; field ; field ; field ; .*

OPTIONS**GENERAL OPTIONS****LOG_HANDLEROUTPUT**

Set to True to log the output of the startup handlers to the logfile.

Example: LOG_HANDLEROUTPUT=False

Default: LOG_HANDLEROUTPUT=False

SCREEN_HANDLEROUTPUT

Set to True to print the output of the startup handlers to the screen.

Example: SCREEN_HANDLEROUTPUT=False

Default: SCREEN_HANDLEROUTPUT=False

LOG Log output dir of edrcinit. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: LOG=/var/opt/ACME/log

Default: LOG=/var/log

CHECK_INTERVAL

Interval of edrcinit checks in seconds.

Example: CHECK_INTERVAL=600,120

Default: CHECK_INTERVAL=600,120

WATCHDOG_NAME

Name of the watchdog that guards the services. This setting has to be changed only if there is a clash with another watchdog using the same name on the system.

Example: WATCHDOG_NAME=EDRCINIT

Default: WATCHDOG_NAME=EDRCINIT

EDRC_USER

User in which HOME WA2L/edrc is installed.

Example: EDRC_USER=edrc

Default: EDRC_USER=edrc

SERVICE HANDLING OPTIONS

The format of the service handling specification is:

sequence ; activated ; guarded ; name ; description ; handler ; handleroptions ;

where the fields have the following content:

sequence startup sequence number.

activated startup activated (**Y**=activated, **N**=not activated).

guarded service monitored and guarded (**Y**=YES, **N**=NO).

name name of the service (should be unique).

description
short service description.

handler service handler

handleroptions
optional service handler options.

EXAMPLE

```
#
# etc/edrcinit.cfg - configuration file for edrcinit
#
# [00] 26.05.2009 CWa    Initial Version
#

##
## SERVICES HANDLING OPTIONS
##

# Definition of services to be started/stopped.
#
# Format:
#
#    <sequence>;<activated>;<guarded>;<name>;<description>;<handler>;<handleroption.
#
# Fields:
#
#    <sequence>            = startup sequence
#    <activated>          = startup activated (Y=activated, N=not activated)
#    <guarded>            = service monitored and guarded (Y=YES, N=NO activated)
#    <name>                = name of the service (should be unique)
#    <description>        = short service description
#    <handler>            = service handler
#    <handleroptions>     = optional service handler options
#
100;Y;N;logcheckd        ;log check and report daemon        ;WA2Ledrc.logcheckd;;
110;N;Y;passwdsyncd      ;password synchronization daemon ;WA2Ledrc.passwdsyncd;;
120;N;Y;tthttpd_doc      ;HTTP server (WA2L/edrc:doc)     ;WA2Ledrc.tthttpd;doc;
121;Y;Y;nginx_doc        ;HTTP server (WA2L/edrc:doc)     ;WA2Ledrc.nginx;etc/nginx
130;N;Y;tthttpd_base     ;HTTP server (WA2L/edrc:base)    ;WA2Ledrc.tthttpd;base;
140;Y;Y;tthttpd_foswiki   ;HTTP server (WA2L/edrc:foswiki) ;WA2Ledrc.foswiki;current
150;Y;Y;tthttpd_report   ;HTTP server (WA2L/edrc:report) ;WA2Ledrc.tthttpd;report;
190;Y;Y;shellinaboxd     ;command line shell thru web     ;WA2Ledrc.shellinaboxd;;

##
## GENERAL OPTIONS
##

#
# Fileformat:
#
#    OPTION=<VALUE>
#
```

```
# Set to True to log the output of the startup handlers to the logfile.
#
LOG_HANDLEROUTPUT=False

# Set to True to print the output of the startup handlers to the screen.
#
SCREEN_HANDLEROUTPUT=False

# Log output dir of edrcinit. If you specify a relative path name the
# path is relative to the root of the EDRC installation.
#
LOG=var/log

# Interval of edrcinit checks in seconds.
#
CHECK_INTERVAL=600,120

# Name of the watchdog that guards the services. This setting has to
# be changed only if there is a clash with another watchdog using the
# same name on the system.
#
WATCHDOG_NAME=EDRCINIT

# User in which HOME WA2L/edrc is installed.
#
EDRC_USER=edrc
```

SEE ALSO

edrcintro(1), **edrcinit(1m)**, **edrcinit.handler(3)**, **watchdog(1)**

NOTES

-

BUGS

-

AUTHOR

edrcinit.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcinit.handler – handle start/stop/refresh/status of a service

SYNOPSIS

edrc/lib/edrcinit/PreFix.ApplicationName

AVAILABILITY

WA2L/edrc

DESCRIPTION

handler to start/stop/status/refresh services for an application handled by the **edrcinit**(1m) command.

OPTIONS

- | | |
|----------------|---|
| start | start the service. |
| stop | stop the service. |
| refresh | reload the service configuration (if possible without service interruption). |
| status | return 0 if the service is running (UP), return 1 if the service is not running (DOWN). |

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---|
| 0 | handler could execute the related action with success. The status of the service is: UP. |
| 1 | handler executed the related action with failure. The status of the service is: DOWN. |

FILES

edrc/lib/edrcinit/

directory where all **edrcinit**(1m) handlers are saved.

EXAMPLES

See in directory **edrc/lib/edrcinit/** for example handler scripts.

SEE ALSO

edrcintro(1), **edrcinit**(1m)

NOTES

-

BUGS

-

AUTHOR

edrcinit.handler was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcenv – display the environment variables exported by edrc

SYNOPSIS

edrc/bin/edrcenv

AVAILABILITY

WA2L/edrc

DESCRIPTION

Print all environment variables that are exported to the recovery scripts and the shell by **edrc**. Aside the environment variable names the current values are printed, too.

Do not use other variables seen by your scripts which are related to the WA2L/edrc environment because they may disappear in future releases or its content may change without notice.

The **edrcenv** is intended to be used in the **shell** invoked within **edrc** to quickly display all official **edrc** environment variables.

OPTIONS

-

ENVIRONMENT

For an explanation of all environment variables listed with **edrcenv** see section **ENVIRONMENT** in **edrc(1m)**.

FILES

-

SEE ALSO

edrc(1m), **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

edrcenv was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrclicense – GNU General Public License for WA2L/edrc

SYNOPSIS

edrc/doc/COPYING

edrc/sbin/edrc → license

AVAILABILITY

WA2L/edrc

DESCRIPTION

To display the WA2L/edrc license, invoke the **license** contributed command after the start of **edrc(1m)**, the **Enterprise Disaster Recovery Console**.

WA2L/edrc LICENSE

WA2L/edrc LICENSE
Version 1, January 2003

Copyright © 2003 – 2025 Christian Walther

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

A copy of the GNU General Public License is distributed with WA2L/edrc and is saved in edrc/doc/COPYING .

GNU GENERAL PUBLIC LICENSE

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the

program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a

notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include

anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be

mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

FILES

edrc/doc/COPYING

GNU General Public License.

SEE ALSO

edrcintro(1), **contrib.license(1m)**, <https://www.gnu.org/licenses/gpl-2.0.html>

NOTES

-

AUTHOR

edrclicense was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcman – display WA2L/edrc manual pages

SYNOPSIS

edrc/bin/edrcman [**-h** | **-w** | **-s** | **-m** | **-l** ([*os*] | *sys*)]

edrcman [*manpage*]

edrcman [*section*] *manpage*
edrcman *manpage* [. *section*]

edrcman [*man_options*] [*section*] *manpage*
edrcman [*man_options*] *manpage* [. *section*]

edrcman -o [*OS*] [*man_options*] [*section*] *manpage*
edrcman -o [*OS*] [*man_options*] *manpage* [. *section*]

edrcman -i [*SYSTEM*] *manpage* [. *section*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **edrcman** you can view the manual pages provided with WA2L/edrc without setting the `$MANPATH` environment variable. To display the manpages the operating system's own **man** command is used internally.

OPTIONS

- h** show **edrcman** usage.
- help** show **man**(1) usage.
- w** print `$MANPATH` that is searched by the **man** command.
- s** print `$MANSECT` that defines the manual page sections that are searched by the **man** command.

-m list man page mappings from **edrc/etc/edrcman.map**.

-l ([*os*] | *sys*)
list available bundled *OS* or *SYSTEM* man pages.

-o [*OS*] show/search only the WA2L/edrc bundled native manual pages of the related operating system.

The *OS* (not case sensitive) can be an operating system ID as returned by **osid(3)**, or the operating system name and version (use: **-l** to list the possible *OS* specifications).

If the *OS* is not specified, the **\$MAN_OSID** is used as *OS*.

-i [*SYSTEM*]
show internet based (<https://manned.org/>, <https://man.freebsd.org/>) manual pages of the given operating *SYSTEM* variant.

Available *SYSTEM*s are: **alpine**, **arch**, **centos**, **darwin**, **debian**, **dragonfly**, **fedora**, **freebsd**, **hpux**, **irix**, **linux**, **macos**, **macosx**, **minix**, **netbsd**, **openbsd**, **opendarwin**, **opensuse**, **osf1**, **plan9**, **rocky**, **slackware**, **solaris**, **sunos**, **sunos4**, **sunos5**, **suse**, **true64**, **ubuntu**, **ultrix**, **v7**, **x11**, **xfree86** and to query the general manual pages: **[M]ANNED** or **[F]REEBSD**, whereas the default is derived from the **/etc/os-release** file, if possible.

man_options
all native man options.

manpage manual page to be read. If no *manpage* is specified, the **edrcintro(1)** is displayed.

section manual page section.

ENVIRONMENT

\$MAN_OSID
if this environment variable is not set, the **osid(3)** command. The only reason to set the **\$MAN_OSID** variable is, when it is needed to check manual pages of operating system dependent commands without logging in to a system resolving that operating system to the related OS-id.

\$LANG if **LC_ALL** in **edrcman.cfg** is not set and **\$LANG** is not empty, the contents of **\$LANG** is used as **\$LC_ALL** for the **man** command. Else the *C* locale is used as default.

\$LC_ALL if **LC_ALL** is not defined in **edrcman.cfg** the contents of the environment variable is used for the **man** command

other for other possible environment variable settings see **man(1)**.

EXIT STATUS

- 4** usage displayed.
- 6** config file **edrcman.cfg** not found.
- 16** manual page requested does not exist. If the requested page is a currently missing WA2L/edrc manual page, a message indicating to call **edrcman edrcintro** is printed.
- 17** for the operating system specified in the **-o OS** option no native manual pages are bundled with WA2L/edrc.
- x** the exit status of native **man**, see **man(1)** for more information.

FILES

- edrc/man/** manual pages.
- edrc/man/all**
WA2L/edrc manual pages and manual pages for all operating systems.
- edrc/man/all/man1/EDRC.1**
index man page of WA2L/edrc.
- edrc/man/<OS-id>/**
manpages of operating system dependent WA2L/edrc commands.
- edrc/man/OS/<OS-id>/**
manpages collection of operating system commands. This to provide (closely native) operating system manual pages on systems where the manual pages are not installed or to check command usage of other operating systems (**export MAN_OSID=osid**) when developing commands (see also: **mandir(4)** manpage). To view this manual pages, add


```
'approot'/man/OS/${MAN_OSID}
```

preferably as last entry to the **MAN_PAGE=...** setting in **edrc/etc/edrcman.cfg**.
- edrc/etc/edrcman.cfg**
configuration file of **edrcman**, see **edrcman.cfg(4)** for more information.
- edrc/etc/edrcman.map**
mapfile to map certain manual pages.
- edrc/etc/perl_wrapper.cfg**
configuration file of **perl_wrapper(3)**. The **MODULE_DIR** setting of this file is read to resolve the man page location of perl modules bundled with WA2L/edrc. See **perl_wrapper.cfg(4)** for more information.

edrc/var/cache/edrcman/

resolution cache of the **edrcman** command if the **CACHE** setting in the configuration file **edrc/etc/edrcman.cfg** is not set to *False*.

edrc/var/cache/edrcman/meta

meta data for the resolution cache. This file is used to check if the cache data is still valid.

edrc/var/cache/edrcman/data

this file contains the cache data, the resolved **MANPATH** and **MANSECT** settings for the system where **edrcman** has been executed.

/etc/os-release

Linux operating system release information file.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcman.cfg(4)**, **edrcman.map(4)**, **edrc(1m)**, **man(1)**, **mandir(4)**, **manpages(4)**, **osid(3)**, **perl_wrapper.cfg(4)**

NOTES

If you called the shell command within **edrc** the **man** command is an alias to **edrcman**.

BUGS

-

AUTHOR

edrcman was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcman.cfg – configuration file for edrcman

SYNOPSIS

edrc/etc/edrcman.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **edrcman** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

CACHE Cache settings resolved by **edrcman** if setting is not set to **False**.

Example: **CACHE=False**

Default: **CACHE=True**

LC_ALL Set **LC_ALL** to ensure proper manpage display (also special characters).

Example: **LC_ALL=C**

Default: **LC_ALL=**

MAN_PATH

space separated list of additional man pages which are not available in the **\$MANPATH** of the user starting **edrcman**. This additional manpages will be appended to the standard **\$MANPATH** and are accessible thru the **edrc 'man'** command. Do not add the man-pages of WA2L/edrc to this path, those pages are available by default.

Example: MAN_PATH= /opt/graphics/common/man /opt/hparray/share/man

Default: MAN_PATH= 'approot'/man/<OSID> 'approot'/man/all

MAN_SYS

Operating system variant name.

Available names are: **alpine**, **arch**, **centos**, **darwin**, **debian**, **dragonfly**, **fedora**, **freebsd**, **hpux**, **irix**, **linux**, **macos**, **macosx**, **minix**, **netbsd**, **openbsd**, **opendarwin**, **opensuse**, **osf1**, **plan9**, **rocky**, **slackware**, **solaris**, **sunos**, **sunos4**, **sunos5**, **suse**, **true64**, **ubuntu**, **ultrix**, **v7**, **x11**, **xfree86** and for general manual pages: **[M]ANNED** or **[F]REEBSD**.

If no **MAN_SYS** is set the name is resolved from the **/etc/os-release** configuration file.

Example: MAN_SYS=ALL

Default: MAN_SYS=

SEE ALSO

edrcintro(1), **edrcman**(1)

NOTES

-

BUGS

-

AUTHOR

edrcman.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcman.map – manual page mappings

SYNOPSIS

etc/edrcman.map

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the manual page map file for the **edrcman** command.

The mappings defined here take into effect, when the manual page is not found in other locations of the **\$MANPATH**.

FILEFORMAT

Rows starting with a **#** are considered as comments.

ENTERED;*MAPPED*;

Where:

ENTERED manual page one entered on the command line: **man** *manual-page* .
The *ENTERED* field entry is not case sensitive.

MAPPED this entry specified the manual page that is displayed in place of the one entered on the command line.
The *MAPPED* field entry is not case sensitive.

EXAMPLES

```
#
# edrcman.map - map manual pages
#
# [00] 10.02.2019 CWa   Initial Version
#
FROM;TO;
a;edrcintro;
abc;edrcintro;
history;edrcintro;
intro;edrcintro;
ll;ls;
llcomp;edrcintro;
lscomp;edrcintro;
month;edrcintro;
name;edrcintro;
path;edrcintro;
```

```
pathlist;edrcintro;  
pid;ps;  
revision;edrcintro;
```

SEE ALSO

edrcintro(1), **edrcman(1)**, **edrcman.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

edrcman.map was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcpack – pack WA2L/edrc application to a software package

SYNOPSIS

edrc/bin/edrcpack [**-h**]

edrcpack [**-d** *output_dir*] [**-p** *version*] [**-s**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

!! edrcpack IS DEPRECIATED AND THE FUNCTIONALITY HAS BEEN REDUCED, USE pack !!

edrcpack uses the command **pack -a edrc** *edrcpack_options* internally.

But only a very reduced set of options of **pack** is available when using **edrcpack**. Therefore use **pack** to create a software package of WA2L/edrc.

See **pack**(1m) for the description.

OPTIONS

See **pack**(1m) for the detail description of all options.

-h usage message.

-d *output_dir*
 directory where to put the package file.

-p *version* patch package to the specified version.

-s split package also in pieces which fit to a diskette.

ENVIRONMENT

See **pack**(1m).

EXIT STATUS

See **pack**(1m).

FILES

See **pack**(1m).

EXAMPLES

See **pack**(1m).

SEE ALSO

edrcintro(1), **pack**(1m), **pack.cfg**(4)

NOTES

edrcpack is only available for backward compatibility and no longer maintained.

BUGS

See **pack**(1m).

AUTHOR

edrcpack was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcports – directory of network ports and protocols used by WA2L/edrc

SYNOPSIS

-

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is a directory of network ports used by default by the WA2L/edrc package.

The following table shows the ports needed for a certain WA2L/edrc command/service to be able to run.

Index:

- x = mandatory, needed for base functionality
- + = mandatory, needed on bidirectional connection (between WA2L/edrc systems)
- . = not mandatory, but needed if tools shall be used on bidirectional connection (between WA2L/edrc systems)
- L = localhost connection only.
- ! = default port (can be changed in configuration file or on command line)

Reasonable User Experience: WA2L/edrc

For a reasonable user experience and most common use where WA2L/edrc is installed on multiple nodes within customer environment(s), at least the following protocols and ports should be enabled for use. This enables the use of the base functionality of WA2L/edrc without using advanced capabilities of the toolset:

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|-------|----------|------|---------|----------|--|----------|
| SSH | TCP | 22 | + | + | edrc (and: sat, asup, psup, sys, rsat, lots) | |
| ICMP | IP | - | | | | |
| DNS | TCP/UDP | 53 | | | | |
| HTTPS | TCP | 443 | | x | edrc → edrcupgrade | download |

Details: SSH

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|------|----------|------|---------|----------|---|---|
| SSH | TCP | 22 | + | + | edrc → distribute edrc → edrcupgrade edrc → tail edrc → trunk | also for: sat, asup, psup, sys, lots |
| | | | | | filedist is_up passwdsyncd rcmd remote_copy remote_shell rsat | |
| | | | . | x | loggrep lgrep rcat rcomm rdiff rosid rsync rssh shell <i>host</i> ssh-exec sys2html syspoll whoisin | |

Details: ICMP, SMB

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|--------------|----------|------------|---------|----------|---|---|
| ICMP ECHO | IP | – | + | + | edrc → distribute edrc → edrcupgrade edrc → tail edrc → trunk is_up | also for: sat, asup, psup, sys, lots |
| | | | | | edrc → ping | |
| | | | . | x | traceroute | |
| SMB | TCP | 135 445 | . | x | syspoll winexe wmic | WMI |

Details: RPC

Most likely in your WA2L/edrc environment RPC is not used and not needed. See **edrcsetup(1m)** step 3.1 for additional information.

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|------|----------|------|---------|----------|--|---|
| RPC | TCP/UDP | 111 | + | + | edrc → distribute edrc → edrcupgrade edrc → tail edrc → trunk | also for: sat, asup, psup, sys, lots |
| | | | | | filedist passwdsyncd rcmd remote_copy remote_shell rsat shell <i>host</i> syspoll | |

Details: HTTP

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|------|----------|--------------|---------|----------|--|--------------------------|
| HTTP | TCP | 80 | | x | edrc → leo leo mediawiki | |
| | | ! 80 | . | x | bget lynx wget | |
| | | | . | | tthttpd | |
| | | ! 8802 | x | | WA2L/edrc:base tthttpd | testing |
| | | ! 8888 | x | | WA2L/edrc:doc tthttpd | |
| | | ! 9900 | x | | WA2L/edrc:report tthttpd | |
| | | ! 9902 | x | | WA2L/edrc:edrcapi tthttpd | |
| | | ! 8806 | x | | WA2L/edrc:shellinaboxd shellinaboxd | |
| | | 8082 9092 | L | | h2 | local:8082 local:9092 |
| | | many | . | x | connect | |

Details: HTTPS

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|-------|----------|--------|---------|----------|---|------------|
| HTTPS | TCP | 443 | | x | edrc → edrcupgrade edrc → leo leo | download |
| | | ! 443 | . | x | lynx wget | |
| | | | . | | nginx | |
| | | ! 8889 | x | | WA2L/edrc:doc nginx | proxy 8888 |
| | | ! 9901 | x | | WA2L/edrc:report nginx | proxy 9900 |
| | | ! 9903 | x | | WA2L/edrc:edrcapi nginx | proxy 9902 |
| | | ! 8807 | x | | WA2L/edrc:shellinaboxd nginx | proxy 8806 |

Details: DNS, SOCKS, SSL, OTHERS

| NAME | PROTOCOL | PORT | INBOUND | OUTBOUND | COMMAND / SERVICE | COMMENTS |
|-----------|----------|-------|---------|----------|---|-------------------------------------|
| SMTP | TCP | 25 | | x | mail_file msmtp logcheckd | |
| | | | | | edrc → logviewer edrc → scriptpack edrc → scriptdocbook | mail file mail file mail file |
| | | | | | | |
| DNS | TCP/UDP | 53 | | x | dig edrc → ping fssum hwinventory hostaliases is_running lots mail_file syspoll | |
| | | | | | edrc → distribute edrc → trunk filedist rcmd | |
| SSL | TCP | ! 990 | . | x | ftps | |
| SOCKS 4/5 | TCP | 1080 | . | x | connect | |
| many | TCP/UDP | many | . | x | nc | |
| many | many | many | . | x | nping | |

FILEFORMAT

-

OPTIONS

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **binprobe(1m)**, **protocols(5)**, **services(5)**, **<http://www.iana.org/assignments/port-numbers>**

NOTES

-

BUGS

-

AUTHOR

edrcports was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcperm.no_shell – pseudo shell to deny access to a shell from vi, more, etc.

SYNOPSIS

edrc/lib/edrc/edrcperm.no_shell

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **edrcperm.no_shell** command is used within **edrc** as a pseudo shell to deny access to the real shell via the built in commands of **vi**, **view** and **more** when the *shell* permission is listed in **DENY_LIST** in the **edrc** config file **etc/edrc.cfg**.

Beside the stdout message, a log entry is created.

The following dialog appears, if a user tries to invoke the shell within **vi** :

```
edrc-WARNING execution of functionality 'shell' not permitted.  
Press RETURN to continue
```

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

EXAMPLES

1) How to implement shell access denial

Code cut-out to deny shell access within commands that provide shell access as **vi**, **view** and **more** :

```
:  
save_shell=$SHELL  
export SHELL='aproot `/lib/edrc/edrcperm.no_shell
```

```
vi $file
export SHELL=$save_shell
:
```

SEE ALSO

edrcintro(1), **edrc(1m)**

NOTES

-

BUGS

-

AUTHOR

edrcperm.no_shell was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcrevision.cfg – configuration file for edrcrevision

SYNOPSIS

edrcrevision/etc/edrcrevision.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file containing the revision of the WA2L/edrc package.

edrc reads the **edrcrevision.cfg** file to resolve the package revision. To display the package revision invoke **edrc -V** or **banner** within **edrc**.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**VERSION**

Version of the WA2L/edrc in the format <major>.<update> . See **edrcrevision(1)** section **RELEASE-NUMBERS** for an explanation of the release numbering system used in WA2L/edrc.

Example: VERSION=1.5

Default: -

PATCHLEVEL

Patchlevel of the WA2L/edrc in the format <patchlevel> . The Patchlevel is a two-digit number. See **edrcrevision(1)** section **RELEASE-NUMBERS** for an explanation of the release numbering system used in WA2L/edrc.

Example: PATCHLEVEL=02

Default: -

SEE ALSO

edrcintro(1), **apprevision(3)**, **edrcrevision(1)**, **edrc(1m)**

NOTES

-

BUGS

-

AUTHOR

edrcrevision.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcroot – print the root of the WA2L/edrc installation

SYNOPSIS

edrc/lib/edrcroot

AVAILABILITY

WA2L/edrc

DESCRIPTION

!! DO NOT USE edrcroot FOR NEW COMMANDS, USE approot !!

With **edrcroot** the root of the WA2L/edrc software installation is printed. This command is used inside the command to evaluate the basedir.

EXIT STATUS

0 always

SEE ALSO

edrcintro(1), **approot(3)**

NOTES

edrcroot will be replaced by **approot** completely as soon as all commands has been migrated to use **approot** instead of **edrcroot**.

BUGS

-

AUTHOR

edrcroot was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcscripts – writing recovery scripts in edrc

SYNOPSIS

edrc/bin/scripts/CUSTOMER-edrc/...

edrc/bin/scripts/CUSTOMER-sat/...

edrc/bin/scripts/CUSTOMER-sat/sup:Support/osup:OperationSupport/...

edrc/bin/scripts/CUSTOMER-sat/sup:Support/psup:ProductionSupport/...

CUSTOMER ::= ‘server_environment -C’

AVAILABILITY

WA2L/edrc

DESCRIPTION**READ ALSO**

It is highly recommended to also read the manual pages listed in the **SEE ALSO** section, especially **edrc(1m)** and **edrcintro(1)** because the WA2L/edrc package provides many commands and concepts that significantly ease up the efficient development of recovery scripts.

CREATE/RENAME/REMOVE RECOVERY SCRIPTS

Recovery scripts are created using the **newscript** command in **edrc (1m)**, are renamed using the **rename-script** command and are removed using **removescript**.

The **newscript** command provides a script file template and ensures that the recovery script name adheres to the supported name convention.

edrc maintains an environment that makes it obsolete to initialize common settings of the operating system in each script and makes a recovery script tree portable.

CENTRAL INITIALIZATION AND COMMAND EXECUTION

If it is needed to centrally initialize variables or execute commands before each script in a menu, the **env** command in **edrc** supports to do this initialization or command start before each script by handling an environment-file that sits in the menu.

Candidates of commands to be executed in an environment-file created with **env** are for example **scripttitle(3)** and **scriptsequence(3)**.

CREATE/RENAME/REMOVE (SUB) MENUS

(Sub)Menus are created, renamed and removed using the **newmenu**, **renamemenu** and **removemenu** commands in **edrc**.

DOCUMENTATION

A recovery script can easily be documented using **doc** tags as understood by the **contrib.doc (1m)** command.

Doing this the description can be easily displayed using the **doc** contributed command, which generates an ah-hoc manual page where the essentials to know about a script, as if a RE-RUN of the script is allowed, the duration of the execution or an example output and more can be documented.

This documentation can be used later to produce a whole documentation, the ScriptDocBook using the **scriptdocbook** command that can be used for offline-reading in a Web-browser without the need to write another documentation.

OPTIONS

recovery scripts do not receive options, all input needs to be queried, best using the **input(3)** and **choice(3)** commands which are tuned to work smooth with **edrc(1m)**.

ENVIRONMENT

See section **ENVIRONMENT** in **edrc(1m)** for a description of all environment variables exported to a recovery script.

Use the **edrcenv(1)** command in the **shell** started from **edrc** or the **edrcenv** contributed command in **edrc** to print the official environment variables that can be used.

EXIT STATUS

a recovery script can exit with any exit status, it will not be resolved.

FILES**\$EDRC_TMP_DIR/**

secure temporary directory that should be used by a recovery script to save temporary data.

The directory with a unique name is created new just before the start of a recovery script and is removed after the script execution.

Therefore a recovery script doesn't need to take care of the temporary directory.

\$EDRC_TMP_DIR_SESSION/

this variable points to a secure temporary directory that exists during the execution of an **edrc** session and is afterwards removed. This variable is exported to recovery scripts, contributed commands and the shell command only.

This variable should be used to securely save temporary data from recovery scripts or contributed commands that must be accessed multiple times throughout the **edrc** session.

You should add the name of the recovery script or contributed command as prefix to ensure a unique filename.

Example:

```
date > $EDRC_TMP_DIR_SESSION/$EDRC_SCRIPTNAME.data
```

If you need to save temporary data during the execution of a recovery script or a contributed command the **\$EDRC_TMP_DIR** variable shall be used.

EXAMPLES

1) Example environment (_env) file:

```
#
# _env - Environment settings for commands in /DR
#
# [00] 07.05.2020 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

scripttitle
scriptsequence -a query || exit

export AVAMAR_ADMIN="admin"
export AVAMAR_SYSADMIN="sysadmin"

export CONNECTION_USER="edrc"
```

2) Example recovery script:

```
#!/bin/bash
#
# 1291:setup_ndmp - setup NDMP accelerator
#
# [00] 06.05.2020 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

# const and defaults
#
user_at_node=root@acme-007.acme.ch
root_password="\pwsafe root@IDPA\""

# query information
#
while [ "$ok" != "y" ]; do
    echo
```

```

        root_password=`input "Root server password" "$root_password" \
                           LOG_STARS,NOT_NULL,NO_TRIM`
        echo
        ok=`choice "input OK? <yn>" yn n`
done
echo

# execute commands
#
ssh-exec -l $CONNECTION_USER -u $user_at_node <<EOM

[COMMANDS]

avsetupndmp

[ANSWERS]

;avsetupndmp will stop all running agents;Y\r;
;Network address or DNS name of this Accelerator Node;\r;
;Network address or DNS name of the Avamar Server;\r;
;Root password of Avamar Server;$root_password\r;
;Repeat password;$root_password\r;
;Enable support for multiple simultaneous backups Y/N?;Y\r;
;Enter action;;5\r;

EOM

# D: Setup NDMP accelerator nodes:
# D:
# D: - setup NDMP accelerator node bkpviewbad01
#
# R: YES
#
# O: Root server password [*****]: *****
# O:
# O: input OK? <yn>  [n] :y
# O:
# O: avsetupndmp version 5.0
# O: Configure systems for backup and restore.
# O: avsetupndmp will stop all running agents, do you wish to continue Y/
# O:
# O: avagent Info: -----
# O: avagent Info: Client 'acme-008-0815' in directory '/usr/local/avamar
# O: avagent Info: Client Agent acme-008-0815 not running.
# O: avagent Info: -----
:
:
# O: avagent Info: -----
# O:
# O: IMPORTANT: Be sure to run avregister if you have added new accounts.
#
# N: Chapter: 6.16, Step 1
#

```

```
# T: ~10 seconds
#
```

3) Examples in script trees part of the WA2L/edrc package:

See "recovery" script examples in the **sys (1)** maintenance configuration located in **edrc/scripts/sys/**.

SEE ALSO

edrcintro(1), **edrc(1m)**, **approot(3)**, **choice(3)**, **contrib(1m)** **contrib.edrc(1m)**, **contrib.doc(1m)**, **contrib.pwsafe(1m)**, **pwsafe(3)**, **contrib.scriptsequence(1m)**, **scriptsequence(3)**, **hostlist(3)**, **hostlist.dat(4)**, **input(3)**, **input_targets(3)**, **outex(1)**, **rel2abs(3)**, **scripttitle(3)**, **server_environment(3)**, **ssh-exec(1)**

NOTES

-

BUGS

-

AUTHOR

edrcscripts was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

edrcsetup – How to install WA2L/edrc (Quick Installation Guide)

AVAILABILITY

WA2L/edrc

DESCRIPTION

The following seven steps explain how to efficiently perform an initial setup of WA2L/edrc in an environment.

The distribution package is a shell archive (example: **edrc-1.4.08-200502150919.sh**). When you execute the shell archive all needed credentials are prompted and the installation is straight forward.

If you received WA2L/edrc as a splitted archive (example: **edrc-1.0.00-200401281209.sh.piece_aa ... ax, edrc-1.0.00-200401281209.sh.piece.sh**) you have to invoke the **edrc-1.0.00-200401281209.sh.piece.sh** command first to concatenate the pieces to the shell archive (see step 0 below).

REQUIREMENTS

The WA2L/edrc package has the following requirements to software installed on the system:

Bourne Shell

the Bourne shell **/bin/sh** has to be installed in **/bin/**.

Korn- or Bourne-Again Shell

The Korn shell **ksh** or the Bourne-Again shell **bash** has to be installed somewhere on the system. See **ksh_wrapper(1)** for additional information.

Perl

The Perl interpreter **perl** has to be installed somewhere on the system. See **perl_wrapper(1)** for additional information.

AWK

The AWK interpreter **awk** and **nawk** (only on Solaris) has to be installed somewhere in the standard **\$PATH** of the related operating system.

Java

The Java runtime environment **java** should be installed somewhere on the system. See **java_wrapper(1)** for additional information. If Java is not installed only a limited non-mandatory set of commands of WA2L/edrc is not available. See **compatibility(1)** and watch out for *JBIN* entries.

INSTALL STEPS

For a productive installation which will profit from the full functionality and is used for disaster recovery (DR) and/or operating system administration WA2L/edrc should be installed as user *root*.

0. Concatenate the splited WA2L/edrc package

If you received the archive splited into pieces, you have to concatenate the pieces first. A splited archive consists of a setup script and the splited archive: **edrc-1.0.00-200401281209.sh.piece.sh** (command to concatenate the pieces) **edrc-1.0.00-200401281209.sh.piece_ab ... ax** (splited archive). The **edrc-1.0.00-200401281209.sh.piece.sh** takes care that you have all needed pieces available and that those pieces are concatenated correctly.

```
host-001# sh ./edrc-1.0.00-200401281209.sh.piece.sh
```

→ goto step 1.

1. Copy the WA2L/edrc package to the first host

Ensure that you transfer the package in ***BINARY*** mode to the target system. Many SCP or SFTP clients on Windows (as FileZilla) will transfer this package in ASCII mode if the transfer mode is not explicitly specified as BINARY.

```
Start FileZilla
-> Connect to server using: SFTP - SSH File Transfer Protocol
-> Set Standard Transfer Type to: Binary
-> Transfer the package file to: host-001:/tmp
```

→ goto step 2.

2. Install the WA2L/edrc software

2.1 Execute the shell archive to install the software

Only install the software on the initial system.

In the subsequent steps the installation is configured and repacked. Then the repacked software is installed on all other hosts in the environment.

```
host-001# sh /tmp/edrc-1.4.08-200502150919.sh
```

The install script will prompt for all credentials and create the *edrc* user/group if selected:

```
***
***
***      SOFTWARE SHELL ARCHIVE
***
***      E D R C - Enterprise Disaster Recovery Console
***      UNIX/Linux version 1.4
```

```

***      patchlevel 08
***      developed on Solaris 8, HP-UX 11.00, redhat Linux 7.2/8.0
***      packed Mon Feb 15 09:19:54 MET 2005
***
***      Copyright(C) 2003-2005
***      Christian Walther
***
***      Send comments and bugreports to <wa2l@users.sourceforge.net>
***
Installation Logfile ..... : /tmp/edrc-1.4.08-200502150919.sh.1291.1

Only extract .cpio.gz ..... <yn> [n]:
Installation Base Directory ... [/opt]:
Verbose File Extraction .... <yn> [n]:
Create Software User ..... <yn> [n]: y
  Username ..... [edrc]:
  User-ID ..... [911]:
  Groupname ..... [edrc]:
  Group-ID ..... [911]:

Input OK? <ync> [n]: y

Start installation to '/opt/edrc'? <yn> [n]: y

install software package ...
  extract software to installation directory ...
    252519 blocks
  done.
  create group ...
    Groupname .. : edrc
    Group-ID ... : 911
  done.
  create user ...
    Username ... : edrc
    User-ID .... : 911
  done.
  check OSID mapping ...
    osid on this system is 'Linux'
  done.
  post installation steps ...
    1) read manpage: /opt/edrc/bin/edrcman edrcsetup
    2) read manpage: /opt/edrc/bin/edrcman edrc
    3) read manpage: /opt/edrc/bin/edrcman edrcintro
  done.
done.

```

HINTS:

- the "Installation Logfile" (**/tmp/edrc-1.4.08-200502150919.sh.1291.log**) content is appended to the install logfile (**edrc/var/log/install.log**) in the related installation directory when the installation of the WA2L/edrc package was possible.
- the temporary directory **/tmp/** has to have sufficient disk space holding at least the size of the shell archive. If this temporary space is not sufficient, the installation will abort and a message is printed. To

be able to install the WA2L/edrc package on systems with low free space in the **/tmp/** directory, set the **\$TMPDIR** variable to a directory having sufficient free space. Ensure that the directory specified in **\$TMPDIR** exists and that the install user has write permission in it.

- if the installation directory has not sufficient space to install the WA2L/edrc package, the installation will abort and a message is printed.

→ goto step 2.2

2.2 Edit the **osid.dat.WA2L** file to ensure that your operating system is supported.

WA2L/edrc has been developed on HP-UX 11.00 and tested on 11.11 ... 11.31, Solaris 8 ... 10 SPARC and x86, RedHat Linux 7.2, 8.0, Fedora Core 4 ... 17, SuSE Linux 5.3 ... 11.1, Ubuntu 14.04 and has been run on other flavors.

The installation gives the hint in

```
:
check OSID mapping ...
    osid on this system is 'Linux'
done.
:
```

if the operating system id (OSID) could be resolved.

If the OSID cannot be resolved, the **osid** command returns **unknown**:

```
host-001# ~edrc/lib/osid

unknown
```

The installation of the shell archive will give a hint of the OSID definition, as:

```
Linux:3.6.11-5.fc17.x86_64:x86_64:unknown:<Comment>:
```

The *unknown* placeholder must be replaced by a matching OSID, the *<Comment>* placeholder should be replaced by a meaningful human readable operating system description.

The OSID definition line can also be resolved later, invoking the **~edrc/lib/osid -e** command.

If your Operating System is sufficient similar to:

- **HP-UX 11.00, 11.11**, map it to the HP-11 OSID
- **Solaris 8**, map it to the Solaris OSID
- **RedHat Linux 7.2, 8.0**, map it to the Linux OSID
- **non of them**, ask the author about porting plans

See also manpages **osid(3)** and **osid.dat(4)**.

```
host-001# vi ~edrc/etc/osid.dat.WA2L
```

```

1  #
2  # osid.dat.WA2L - operating system map
3  #
4  # [00] ???.???.19?? SFI   Initial Version
5  # [01] 31.01.2003 CWa   Integrated into EDRC
6  #
7  :
8  30 #
9  31 # Fileformat: (see also man osid.dat)
10 32 #
11 33 #   <OS>:<release>:<machine>:<OSID>:<Comment>
12 34 #
13 35 #   <OS>          ::= uname -s
14 36 #   <release>     ::= uname -r
15 37 #   <machine>     ::= uname -m
16 38 #
17  :
18 94 Linux:2.6.18-164.el5:x86_64:Linux:RedHat ELinux Srv 5.4 (Tikanga):
19 95 Linux:2.6.18-164.el5:ia64:Linux-ia:RedHat ELinux Srv 5.4 (Tikanga):
20 96 Linux:2.6.35.[0-9].[0-9]-[0-9].[0-9].fc14.i686:i686:Linux:Fedora Core 14:
21 97 Linux:3.6.11-5.fc17.x86_64:x86_64:Linux:Fedora Core 17:
22 ~
23 ~

```

→ goto step 2.3

2.3 Start the interactive shell

The **shell**(1) command provides a proper environment with all needed **\$PATH** and other environment settings and useful additional commands and functions. Invoke **usage** within the **shell** to get a quick overview.

```

host-001# ~edrc/bin/shell

[ /root ]
[ root@host-001 ][*eshell*/bash]:

```

→ goto step 3.

3. Security setup (rtools and/or secure shell)

If you like to use the **distribute** command of **edrc** (highly recommended), the **~edrc/bin/filedist**, **~edrc/bin/rcmd** commands (very useful) and the **~edrc/bin/passwdsyncd**, edit the corresponding security files. It is possible to run the base functionality of WA2L/edrc in a rtools, secure shell or in a mixed setup.

The newer commands **~edrc/bin/rcat**, **~edrc/bin/rcomm**, **~edrc/bin/rdiff**, **~edrc/bin/rosid**, **~edrc/bin/ssh-exec**, **~edrc/bin/loggrep** and **~edrc/bin/whoisin** need a secure shell setup.

→ goto step 3.2 (for secure shell connection setup)

- goto step 3.1 (for rtools connection setup)
- goto step 3.1 and 3.2 (for a mixed setup)

3.1 Connection using rtools: rsh and rcp (DIST_MODE=rtools)

In nowadays rtools should not be used any more if there is not a really good reason to do so. Instead secure shell as explained in step 3.2. should be used.

- goto step 3.1.1

3.1.1 Edit the `~root/.rhosts` file on all hosts:

```
[ /root ]
[ root@host-001 ][*eshell*/bash]: vi ~root/.rhosts

1 host-001.acme.net edrc
2 host-002.acme.net edrc
:
:
~
~
```

- goto step 3.2 or 4.

3.2 Connection using secure shell: ssh and scp (DIST_MODE=OpenSSH)

Without doubt, this should be the default method to connect to the systems.

- goto step 3.2.1

3.2.1 Create the authentication keys (RSA and DSA)

This are the "local" ssh keys hat are used to connect within one customer environment.

```
[ / ]
[ root@host-001 ][*eshell*/bash]: su edrc

[ / ]
[ root@host-001 ][*eshell*/bash]: id

uid=911(edrc) gid=911(edrc) groups=911(edrc)

[ / ]
[ root@host-001 ][*eshell*/bash]: ssh-keygen -t dsa -C edrc@ACME \
    -f ~edrc/var/connection/security/edrc/OpenSSH/default/default/id_dsa
```

```
[ / ]
[ root@host-001 ][*eshell*/bash]: ssh-keygen -t rsa -C edrc@ACME \
    -f ~edrc/var/connection/security/edrc/OpenSSH/default/default/id_rsa

[ / ]
[ root@host-001 ][*eshell*/bash]: exit
```

See **edrc(1m)**, **remote_copy(3)** and **remote_shell(3)** for more information about the **edrc/var/connection** directory and **ssh-keygen(1)** for more information about authentication key generation.

If you get the error message "open /opt/edrc/.../default/id_dsa failed: Permission denied" can this be related to SELinux (Security-Enhanced-Linux). To fix this issue without to disable SELinux, invoke:

```
[ /opt/edrc/var/connection/security/edrc/OpenSSH/default ]
[ root@host-001 ][*eshell*/bash]: chcon -t ssh_home_t default
```

→ goto step 3.2.2

3.2.2 Add global authentication keys to the ~edrc/etc/ssh-keyadd.pub file

The "global" ssh public keys are those keys that might be used to manage all your customers environments and to connect from your central administration environment.

If you strictly use "local" ssh keys to connect within one customer and the "global" keys to connect from the central admin environment to all your customers and ensure that the private keys ***never*** leave an environment you can create an easy to manage and secure administration situation where you know all your keys.

In this example **ACME** is the customer environment and **Highlander** is the central administration environment (that can consist of a number of Linux/HP-UX/Solaris systems needed to centrally administer all your customers).

```
[ /root ]
[ root@host-001 ][*eshell*/bash]: vi ~edrc/etc/ssh-keyadd.pub

1 #
2 # ssh-keyadd.pub - global SSH public keys
3 #
4 # [00] 19.09.2010 CWa Initial Version
5 #
6 ssh-dss AATAB3NzaC1kc3MAZACBANTZUzTJIZFxNiZprstBI
sgXgFwk/OrIvRMUlkJIrJRBqITLdxF8sQ1lWMjhrufv97yd9o
UAnNv0j9YjBYNdT/ZvHn9VadZYZAiOhQ5N+dvX8EbZWbLtEY
Li49Gwfkf54Tzs0tdle65h4II/LUfE5Z83A3xYpXmB3pkjD1s
4sTvAAAAFTD+03fjToscaJk6dmk1G7RZ7ro8lQAAAIbJcsY8v
iqHK+AcoKTZ+58h8cQAAAEAoSqZXwjZJfXPJ2Q0eqMbqeTAX
T3oML5byrxw+fnANUvIkC8nlUDmQUAg3EgmO7cSya1FPXdkp
qefv4vtninFndj8muwsRhZGXgYFwRggvtaV/rtn3VFLWs0MFp
tdk7NZD6/XCpM9JXJCMubIhhFbQ= edrc@Highlander
7 ssh-rsa ATAAB3NzaC1yc2EAAAABIwAAAEAvkdixAj3Jd9Wf
iuTSuxYqaTZT0/xXepHeuBltwIV1LV6oQNRU8wz6EnzqLYqC
l2qeOTTFBVZVJUvQ55ohDhLQ/7SKLF7B1na4JYAONnqLpR/A+
```

```

cFTI9jQrPHudEhPwS80xx65jSCBOMq+ydZtFLLFnd+HiX/5rz
K86QXC7HCcnfXezr+78jcpBgWDpQ== edrc@Highlander
~
~

```

See **ssh-keyadd.pub**(4) for more information about this configuration file.

→ goto step 3.2.3

3.2.3 Add the authentication keys to the `~root/.ssh/authorized_keys` file

The command **ssh-keyadd** allows to efficiently handle the different ssh keys and also to quickly check the applied keys:

```

[ /root ]
[ root@host-001 ][*eshell*/bash]: ssh-keyadd

ssh-keyadd - add SSH public keys to a user, by Chr. Walther

add public keys to user 'root' ...
  list user's public SSH keys ...
    TYPE  KEY  COMMENT
    ----  ---  -
    (0)
done.
list of global keys to add ...
    TYPE  KEY  COMMENT
    ----  ---  -
    ssh-dss  AATAB3NzaC1kc3MA...pM9JXJCMubIhhFbQ=  edrc@Highlander
    ssh-rsa  ATAAB3NzaC1yc2EA...ezr+78jcpBgWDpQ==  edrc@Highlander
    (2)
done.
add global keys? <yn>  [y] :
list of local keys to add ...
    TYPE  KEY  COMMENT
    ----  ---  -
    ssh-dss  AAAAB3NzaC1kc3MA...Q/oCZennu8x67FA==  edrc@ACME
    ssh-rsa  AAAAB3NzaC1yc2EA...rk+O/5IuBE4mmgpJl  edrc@ACME
    (2)
done.
add local keys? <yn>  [y] :
add keys ...
  edit authfile '/root/.ssh/authorized_keys' ... (global) ... (local) ...
done.
list user's public SSH keys ...
    TYPE  KEY  COMMENT
    ----  ---  -
    ssh-dss  AAAAB3NzaC1kc3MA...Q/oCZennu8x67FA==  edrc@ACME
    ssh-rsa  AAAAB3NzaC1yc2EA...rk+O/5IuBE4mmgpJl  edrc@ACME
    ssh-dss  AATAB3NzaC1kc3MA...pM9JXJCMubIhhFbQ=  edrc@Highlander
    ssh-rsa  ATAAB3NzaC1yc2EA...ezr+78jcpBgWDpQ==  edrc@Highlander
    (4)

```

```

done.
adjust SSH daemon configuration? <yn> [y] :
adjust SSH daemon configuration ...
    edit configfile '/etc/ssh/sshd_config' ... done.
done.
send HUP signal to SSH daemon? <yn> [y] :
send HUP signal to SSH daemon ...
    process '/usr/sbin/sshd' with PID '9268' ... done.
done.
done.

```

See **ssh-keyadd(1m)** for more information about the command.

→ goto step 4.

4. Read at least the following man pages

With **edrcman** you can read the system and WA2L/edrc manpages without the need of adjusting the **\$MANPATH** variable.

If **shell** or **edrc** is started, the **man** command is an alias to **edrcman**.

```

[ /root ]
[ root@host-001 ][*eshell*/bash]: edrcman EDRC

[ /root ]
[ root@host-001 ][*eshell*/bash]: edrcman edrcintro

[ /root ]
[ root@host-001 ][*eshell*/bash]: edrcman edrc

```

→ goto step 5.

5. Configure the WA2L/edrc package

5.1 Edit the server_environment.cfg, hostlist.cfg and hostlist.dat files

Change to the configuration directory of WA2L/edrc.

```

[ /root ]
[ root@host-001 ][*eshell*/bash]: cdetc

```

→ goto step 5.1.1

5.1.1 Edit the server_environment.cfg file

In the **server_environment.cfg** it is defined which customer and environment (e.g. Customer=ACME, Environment=DEVELOPMENT) is related to certain servers.

```
[ /opt/edrc/etc ]
[ root@host-001 ][*eshell*/bash]: vi server_environment.cfg

1  #
2  # server_environment.cfg - config for server_environment
3  #
4  # [00] 24.07.2004 CWa Initial Version
5  #
6
7  # Server environment definition.
8  #
9  # If you log on to a system, the <NAME> field of the
10 # first matching <server_regex> will be returned.
11 #
12 :
13 #
14 #
15 #
16 # Format:
17 # <NAME>:<Description>:<server_regex>:<Customer>:
18 #
19 :
20 #
21 #
22 #
23 #
24 #
25 # ACME
26 #
27 PRODUCTION:Datacenter Bern, PRODUCTION Env:host-3[0-9][0-9]:ACME:
28 PREPRODUCTION:Datacenter Boston, Integration Env:host-2[0-9][0-9]:ACME:
29 TEST:Datacenter Plano, Test Env:host-1[0-9][0-9]:ACME:
30 DEVELOPMENT:Datacenter San Jose, Development Env:host-0[0-9][0-9]:ACME:
31
~
~
```

→ goto step 5.1.2

5.1.2 Edit the hostlist.cfg and hostlist.dat files

The hosts listed in the **hostlist.dat** file are returned when the **hostlist** [*options*] command is used. This enables you to write scripts that dynamically only address the hosts that are located in the same environment (e.g. TEST) as where you are starting the script. In other environments (e.g. PRODUCTION) the identical script can be used without modifications, but it will automatically address the hosts in that environment. With this method your script does not need to know about the actual setup of the environment.

Furthermore, to address a certain set of hosts, hostgroups can be defined.

The **@_DIST** hostgroup is "private" and is used to define the targets where the recovery script tree is distributed when invoking the **distribute** command in **edrc**. The **@_DIST** hostgroup should not be used in scripts.

The **@ALL** hostgroup is to address all hosts that exist in all environments of a customer setup (e.g. DEVELOPMENT + TEST + PREPRODUCTION + PRODUCTION). This hostgroup can be used in scripts.

In the following example the **@APP** hostgroup is used to address all application servers. The **@DB** hostgroup contains all database servers in the environment.

Set the **USE_HOSTLIST_DAT** setting in the **hostlist.cfg** file to *True* as in the given example:

```
[ /opt/edrc/etc ]
[ root@host-001 ][*eshell*/bash]: vi hostlist.cfg

1  #
2  # etc/hostlist.cfg - configuration of hostlist
3  #
4  # [00] 16.02.2004 CWa   Initial Version
5  #
6
7  # Set USE_HOSTLIST_DAT to True, if hosts are listed in
8  # the etc/hostlist.dat file, if hosts are defined directly
9  # in etc/hostlist.cfg, set USE_HOSTLIST_DAT to False.
10 #
11 USE_HOSTLIST_DAT=True
:
```

and add the hosts in the **hostlist.dat** file as in the following example:

```
[ /opt/edrc/etc ]
[ root@host-001 ][*eshell*/bash]: vi hostlist.dat

1  #
2  # etc/hostlist.dat - hostlist csv database file
3  #
4  # [00] 12.08.2008 CWa   Initial Version
5  # [58] 16.03.2013 CWa   chg: to hostlistdat2cfg structure
6  #
7
8  #
9  # Format:
10 #
11 #   <CUSTOMER>;<ENVIRONMENT>;<GROUPS>;<OPTIONS>;<HOSTS>
12 #
13 #   To verify the syntax, use: hostlistdat2cfg -a list_dat
14 #   Only correct entries will be listed.
15 #
16 ACME   ;DEVELOPMENT   ;@APP   ;;host-001 host-002;
17 ACME   ;DEVELOPMENT   ;@DB    ;;host-003;
18 ACME   ;TEST          ;        ;;host-101 host-103;
19 ACME   ;PREPRODUCTION ;        ;;host-201 host-203 host-205;
20 ACME   ;PRODUCTION    ;        ;;host-303 host-308 host-309;
~
~
```

The **@_DIST** and **@ALL** host groups don't have to be defined. The **hostlistdat2cfg** command, that is used to generate the **hostlist** configuration, creates those groups automatically.

If the setting **CMAN_ENVIRONMENT=CUSTOMER.NAME** in **hostlist.cfg** is used, more groups are available automatically when calling the **hostlist** command on a host that is part of the environment that is defined as the central administration environment.

To verify your settings use

```
[ /opt/edrc/etc ]
[ root@host-001 ][*eshell*/bash]: hostlist -p
```

to print all available hostgroups (human readable) and

```
[ /opt/edrc/etc ]
[ root@host-001 ][*eshell*/bash]: server_environment -l
```

to list all defined server environments.

See **server_environment(3)**, **server_environment.cfg(4)**, **hostlist(3)**, **hostlist.cfg(4)**, **hostlistdat2cfg(3)** and **hostlist.dat(4)** for more information. Configuration file samples are available in the **edrc/var/samples/hostlist/** directory.

If you installed WA2L/edrc as user *root* and used the *edrc* user to point to the software, as suggested in step 2.1, the main functions of the WA2L/edrc package are configured now.

→ goto step 5.2

5.2 Edit the .cfg files located in ~edrc/etc on one host

Now all other configuration files of WA2L/edrc can be checked and adjusted.

The general concept of the configurations in WA2L/edrc is, that the configuration files can be kept identical on all hosts across environments and customers.

To achieve this, the **server_environment(3)** and **hostlist(3)** commands are used extensively to set configuration values dynamically whenever possible.

```
[ /root ]
[ root@host-001 ][*eshell*/bash]: cdetc

[ /opt/edrc/etc ]
[ root@host-001 ][*eshell*/bash]: vi *.cfg
```

This step can also be performed later. The files then can be edited on one host and distributed to the others easily using the **filedist(1)** command.

→ goto step 5.3

5.3 Optional on low disk space in install directory

If disk space consumption is of concern the *contents* of the directories **edrc/src/** and **edrc/man/OS/** can be removed without impact to the running WA2L/edrc installation.

```
[ /opt/edrc ]
[ root@host-001 ][*eshell*/bash]: rm -rf man/OS/*

[ /opt/edrc ]
[ root@host-001 ][*eshell*/bash]: rm -rf src/*
```

→ goto step 6

6. Installation of WA2L/edrc on all hosts in the Customer Environment

6.1 Create an installable package of the configured software

Now, when the configuration is done, the software is repacked. This will supersede the configuration effort on the other systems within the customer environments.

```
[ /opt/edrc ]
[ root@host-001 ][*eshell*/bash]: pack

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...
  application information ...
    APPLICATION ..... : default
    APPLICATION_PREFIX ... : edrc_ACME
    APPLICATION_NAME ..... : WA2L/edrc
    APPLICATION_RELEASE .. : 1.4.08
    DESCRIPTION ..... : WA2L/edrc complete
  done.
  package information ...
    format ..... : shar
    type ..... : RELEASE
    file ..... : host-001:/tmp/edrc_ACME-1.4.08-20050401231
  done.
  write sadm file ...(/opt/edrc/var/pack/sadm/edrc_ACME-1.4.08-200504012315
  evaluate files to be packed ...(15137 files)... done.
  evaluate properties of files to be packed ... done.
  pack files to package file ...(258374 kB)... done.
done.
```

→ goto step 6.2

6.2 Copy software package to other hosts

Copy the created package to all environments of one customer.

```
[ /opt/edrc ]
[ root@host-001 ][*eshell*/bash]: scp \
```

```

/tmp/edrc_ACME-1.4.08-200504012315.sh root@host-002:/tmp

[ /opt/edrc ]
[ root@host-001 ][*eshell*/bash]: scp \
    /tmp/edrc_ACME-1.4.08-200504012315.sh root@host-003:/tmp

:
:

→ goto step 6.3

```

6.3 Install newly created package on all hosts and apply ssh keys

Initially this is a manual task.

Later the **filedist(1)** and **rcmd(1)** commands can be used to roll-out files and to execute commands easily on many systems.

For common software handling tasks, as patching, package generation etc. a special "recovery" script tree provided thru the **sys(1m)** short start, can be used.

```

host-002# sh /tmp/edrc_ACME-1.4.08-200504012315.sh
host-002# ~edrc/bin/ssh-keyadd -y

host-003# sh /tmp/edrc_ACME-1.4.08-200504012315.sh
host-003# ~edrc/bin/ssh-keyadd -y

:
:

→ goto step 7

```

7. Use EDRC - Enterprise Disaster Recovery Console

The "EDRC - Enterprise Disaster Recovery Console" startup command is **~edrc/sbin/edrc**. The short starts **~edrc/bin/sat**, **~edrc/bin/asup**, **~edrc/bin/psup** and **~edrc/bin/osup** give access to special purpose "recovery" script trees, as system- or application administration tasks.

From now on you can also use the **filedist(1)** command to easily distribute files and **rcmd(1)** to execute command(s) on all hosts in the environment.

To distribute recovery script trees in **edrc(1m)**, **sat(1m)** etc., the **distribute** command in **edrc** can be used.

Patches of the application can be downloaded from <http://sourceforge.net/projects/wa2l-edrc/> or your repository server and installed using the **edrcupgrade** command in **edrc**.

Currently the following pre-configured "recovery" script trees are distributed with WA2L/edrc:

- **~edrc/sbin/edrc** (= **~edrc/sbin/edrc -c edrc.cfg**)

Disaster recovery script tree.

- **~edrc/bin/sys** (= ~edrc/sbin/edrc -c edrc.sys.cfg)

System configuration handling and some automated EDRC tasks as WA2L/edrc roll-out and WA2L/edrc patch installation.

This menu tree is considered to be part of the WA2L/edrc package and is patched, too. Therefore you should not add/remove/change files in this "recovery" script tree, due to the fact that your changes might be lost after an WA2L/edrc update.

- **~edrc/bin/lotsetl** (= ~edrc/sbin/edrc -c edrc.lotsetl.cfg)

Menu tree to efficiently handle **lots**(1m), (long term data save).

This menu is also considered to be part of the WA2L/edrc package and is patched, too. The config files **lotsetl.cfg**(4), **edrc.lotsetl.cfg** and the topmost **_env** file in the menu tree will not be overwritten by patches and are allowed to be changed.

If adjusting the mentioned configuration files is not sufficient for your needs and you like to adjust the menu tree provided thru **lotsetl**, consider to create a copy of it using **edrc/scripts/lotsetl** as template when invoking the **newscripttree** contributed **edrc** command and adjusting the **SCRIPTS_BASEDIR** setting in **edrc/etc/edrc.lotsetl.cfg** to point to the created copy.

- **~edrc/bin/sat** (= ~edrc/sbin/edrc -c edrc.sat.cfg)

System Administration menus to automate common tasks in the environment.

- **~edrc/bin/asup** (= ~edrc/sbin/edrc -c edrc.asup.cfg)

System Administration menu for application support tasks in the environment. Please pay attention to the special configuration for this menu tree, which is part of the **~edrc/bin/sat** tree, but editing and distribution is denied because the idea is to start it by non-root users via **sudo**.

- **~edrc/bin/psup** (= ~edrc/sbin/edrc -c edrc.psup.cfg)

System Administration menu for production support tasks in the environment. Please pay attention to the special configuration for this menu tree, which is part of the **~edrc/bin/sat** tree, but editing and distribution is denied because the idea is to start it by non-root users via **sudo**.

- **~edrc/bin/osup** (= ~edrc/sbin/edrc -c edrc.osup.cfg)

System Administration menus for operation support tasks in the environment. Please pay attention to the special configuration for this menu tree, which is part of the **~edrc/bin/sat** tree, but editing and distribution is denied.

SEE ALSO

EDRC(1), **edrc**(1m), **edrcintro**(1), **edrcports**(4), **binprobe**(1m), **compatibility**(1), **filedist**(1), **hostlist**(3), **hostlistdat2cfg**(3), **java_wrapper**(1), **ksh_wrapper**(1), **loggrep**(1), **osid**(3), **osid.dat**(4), **pack**(1m), **patchinstall**(1m), **pf_wrapper**(1), **perl_wrapper**(1), **rcat**(1), **rcmd**(1), **rcomm**(1), **rdiff**(1), **rosid**(3), **ssh-exec**(1), **sudo**(8), **sudoers**(5), **sys**(1)

NOTES

See **pack(1m)** for instructions how to create an installable shell archive of an already installed WA2L/edrc package.

To start some commands thru **sudo**, the following lines could be added to the **/etc/sudoers** file, using the **visudo** command. To ensure, that the user has to supply the own password to execute the privileged commands, the settings in line 55 and 56 should be commented out or removed from the file.

In the example below the **PASSWD:** setting results in asking for the password of the calling (unprivileged) user prior to command execution as an additional security measurement.

```
[ /etc ]
[ root@host-001 ][*eshell*/bash]: sav sudoers; visudo

1  #
2  # /etc/sudoers - sudo access definition file
3  #
4  # [01] 25.02.2012 CWa  +*_EDRC_* definitions
5  #
6  ##
7  ## Sudoers allows particular users to run various commands
8  ## as the root user, without needing the root password.
9  ##
10 :
11 ## User Aliases
12 User_Alias    ROLE_EDRC_ADM = john, fred
13 User_Alias    ROLE_EDRC_OPS = barney
14 User_Alias    ROLE_EDRC_USR = wilma, betty
15
16 # WA2L/edrc definitions
17 #
18 ROLE_EDRC_ADM ALL = PASSWD: CMD_EDRC_USR, CMD_EDRC_OPS, CMD_EDRC_ADM
19 ROLE_EDRC_OPS ALL = PASSWD: CMD_EDRC_USR, CMD_EDRC_OPS
20 ROLE_EDRC_USR ALL = PASSWD: CMD_EDRC_USR
21 Cmnd_Alias    CMD_EDRC_ADM = /opt/edrc/bin/shell, /opt/edrc/bin/sat,\
22                                     /opt/edrc/sbin/edrc, /opt/edrc/bin/sys
23 Cmnd_Alias    CMD_EDRC_OPS = /opt/edrc/bin/osup, /opt/edrc/bin/lotsctl
24 Cmnd_Alias    CMD_EDRC_USR = /opt/edrc/bin/asup, /opt/edrc/bin/psup
25 :
26 # In the default (unconfigured) configuration, sudo asks for the root
27 # password. This allows use of an ordinary user account for
28 # administration of a fresh installed system. When configuring sudo,
29 # delete the two following lines:
30 #Defaults     targetpw      # specify passwd of target user i.e root
31 #ALL          ALL=(ALL) ALL # only together with 'Defaults targetpw'
32 :
33 ~
34 ~
35 ~
```

See also **pf_wrapper(1)** for a more detailed description of the **sudoers** example above.

As soon as the **sudoers** entries are defined as given above, the commands **asup**, **edrc**, **lotsctl**, **osup**, **psup**, **sat**, **shell** and **sys** can be started by the user being member of the related role with **root** permissions simply

by adding the directory **~edrc/pbin** to the **\$PATH** environment variable (**export PATH=~edrc/pbin:\$PATH**) in the related personal shell startup rc-file. This also works, if **RBAC** for controlling elevated permissions is used.

Therefore when using the **pbin** directory, a user having elevated permissions on those commands does not need to know, if it is needed to use **sudo sat** or **pfexec sat** on a certain system, it is sufficient to call **sat**.

See section **FILES** in **edrcintro(1)** for a description of the **pbin** directory and **pf_wrapper(1)**.

BUGS

-

AUTHOR

edrcsetup was written by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

env – environment configuration file for ini

SYNOPSIS

edrc/etc/env

\$HOME/.myenv

\$BASEDIR/etc/env

\$BASEDIR/etc/env. domainname

\$BASEDIR/etc/env. hostname

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the environment configuration definition for the **ini** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

KEY:VARIABLE=setting

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

KEY application name.

VARIABLE
environment variable.

setting value settings of the *VARIABLE*.

EXAMPLES

```

#
# etc/env - central environment settings for ini
#
# [00] 07.02.2000 CWa   Initial Version
# [01] 06.02.2000 CWa   integrated into edrc
#
#-----
#
# Fileformat:
#
#   <key>:<variable>={<setting> }
#
#
# Reserved variables:
#
#   DESC      : Description
#   MAN       : man path
#
#   PATH      : command path
#   PATH.pre  : command path
#   PATH.post : synonym for PATH
#
#.....

# some applications (installed in SIG, EDS, FSS environment)
#
gv:DESC=GNU GhostView (1.5)
gv:GV_CMD=/apl/gv${OS}/bin/ghostview
gv:MAN=/apl/gv${OS}/man
gv:GV_NEED_INI=gs

gs:DESC=GhostScript (5.1)
gs:GS_BASE=/apl/gs${OS}
gs:MAN=${GS_BASE}/man
gs:PATH.pre=${GS_BASE}/bin
gs:GS_CMD=${GS_BASE}/bin/gs
gs:GS_LIB=${GS_BASE}/share/ghostscript/5.10:${GS_BASE}/fonts/gs_f_other-5.10/fonts
gs:GS_LIB=${GS_LIB}:${GS_BASE}/fonts/gs_f_std-5.10/fonts
gs:LD_LIBRARY_PATH=${GS_BASE}/lib:/usr/openwin/lib
gs:SHLIB_PATH=${LD_LIBRARY_PATH}

gnuplot:DESC=GNU Plot (3.5)
gnuplot:MAN=/apl/gnuplot${OS}/man
gnuplot:PATH=/apl/gnuplot${OS}/bin

gcc:DESC=GCC for HP-10
gcc:GCC_BASE=/opt/gcc
gcc:MAN=${GCC_BASE}/man
gcc:PATH.pre=${GCC_BASE}/bin ${GCC_BASE}/lib/gcc-lib/hppa1.1-hp-hpux10.20/2.95.2
gcc:INFOPATH=${GCC_BASE}/info:

gcc11:DESC=GCC for HP-11

```



```

gcc11:GCC_BASE=/opt/gcc
gcc11:MAN=${GCC_BASE}/man
gcc11:PATH.pre=/opt/binutils/bin ${GCC_BASE}/bin ${GCC_BASE}/lib/gcc-lib/hppa2.0n
gcc11:INFOPATH=${GCC_BASE}/info:

imagemick:DESC=ImageMagick (Image Viewer)
imagemick:MAN=/apl/ImageMagick${OS}/man
imagemick:IMAGICK_CMD=/apl/ImageMagick${OS}/bin/display
imagemick:SHLIB_PATH=/usr/local/lib:/apl/ImageMagick${OS}/lib
imagemick:IMAGICK_NEED_INI=gs

octave:DESC=GNU Octave (Matlab Compatible) (2.0.13)
octave:OCTAVE_BASE=/apl/octave${OS}/usr/local
octave:PATH=${OCTAVE_BASE}/bin /apl/gnuplot${OS}/bin
octave:MAN=${OCTAVE_BASE}/man
octave:INFOPATH=${OCTAVE_BASE}/info
octave:LOADPATH=${OCTAVE_BASE}/libexec/octave/site/oct/sparc-sun-solaris2.6//
octave:LOADPATH=${LOADPATH}:${OCTAVE_BASE}/share/octave/site/m//
octave:LOADPATH=${LOADPATH}:${OCTAVE_BASE}/libexec/octave/2.0.13/oct/sparc-sun-so
octave:LOADPATH=${LOADPATH}:${OCTAVE_BASE}/share/octave/2.0.13/m//:./:
octave:OCTAVE_CMD="${OCTAVE_BASE}/bin/octave"
octave:OCTAVE_CMD_OPT="--info-file ${OCTAVE_BASE}/info/octave.info --path ${LOADP.

tex:DESC=LaTeX for Solaris, V3.14159 (C version 6.1)
tex:PATH=/apl/latex/teTeX/bin/sparc-solaris2.5

# Oracle applications
#

oraps:DESC=Oracle 8.1.6 for PeopleSoft
oraps:ORACLE_BASE=/ora00/app/oracle
oraps:ORACLE_VERSION=8.1.6
oraps:ORACLE_HOME=${ORACLE_BASE}/product/${ORACLE_VERSION}
oraps:ORACLE_OWNER=ora_ps
oraps:PATH=${ORACLE_HOME}/bin
oraps:SHLIB_PATH=${ORACLE_HOME}/lib
oraps:NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
oraps:NLS_DATE_FORMAT=DD-MON-YYYY
oraps:ORA_NLS33=${ORACLE_HOME}/ocommon/nls/admin/data
oraps:TNS_ADMIN=${ORACLE_HOME}/network/admin
oraps:ORADATA_01=/ora01/oradata
oraps:ORADATA_02=/ora02/oradata
oraps:ORADATA_03=/ora03/oradata

oraps9:DESC=Oracle 9.2.0.2 for PeopleSoft
oraps9:ORACLE_BASE=/ora00/app/oracle9
oraps9:ORACLE_VERSION=9.2.0.2
oraps9:ORACLE_HOME=${ORACLE_BASE}/product/${ORACLE_VERSION}
oraps9:ORACLE_OWNER=ora9psof
oraps9:PATH=${ORACLE_HOME}/bin
oraps9:SHLIB_PATH=${ORACLE_HOME}/lib
oraps9:NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1
oraps9:NLS_DATE_FORMAT=DD-MON-YYYY

```

```
oraps9:ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data
oraps9:TNS_ADMIN=$ORACLE_HOME/network/admin
oraps9:ORADATA_01=/ora01_ps/oradata
oraps9:ORADATA_02=/ora02_ps/oradata
oraps9:ORADATA_03=/ora03_ps/oradata
```

SEE ALSO

edrcintro(1), **env**(4), **ini.cfg**(3)

NOTES

-

BUGS

-

AUTHOR

env was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

envpasswdstrip – strip out passwords from `_env` files

SYNOPSIS

`edrc/bin/envpasswdstrip [-h]`

`envpasswdstrip -print | -execute`

AVAILABILITY

WA2L/edrc

DESCRIPTION

strip passwords from all `_env` files below the current directory. Files in the `.sav` directory are not changed.

The purpose of **envpasswdstrip** is to avoid a hand out of the passwords saved in `_env` files of recovery script trees that will be shared with other administrators.

envpasswdstrip replaces the *VALUES* setting in the definition of the format "**export passwd=VALUE**" or "**export password=VALUE**" with *PASSWORD_STRIPED_OUT*.

OPTIONS

- h** usage message.
- print** do not actually change the `_env` files found starting in the current directory, print only the original and the changed entries of the file.
- execute** change the files `_env` files below the current directory.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 2** operating system not supported yet. See **osid(3)** if you get this error.
- 4** usage printed.

- 11** temporary directory could not be claimed in **/tmp/**. Check the system if you get this error it is an indicator of system intrusion.

FILES

-

EXAMPLES

-

SEE ALSO

edrc(1m), **edrcintro**(1)

NOTES

-

BUGS

-

AUTHOR

envpasswdstrip was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

epub2pdf – convert an ePUB eBook to PDF

SYNOPSIS

edrc/bin/epub2pdf [*file.epub*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

epub2pdf is a command-line tool that quickly generates PDF files from EPUB ebooks. It allows the user to specify page size, fonts, margins, and default paragraph alignment. See also <http://epub2pdf.com/>

OPTIONS

usage message.

file.epub ePUB eBook file to be converted to *file.pdf*

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **pdfmetaedit(1)**, **pdfscissors(1)**

NOTES

-

BUGS

-

AUTHOR

epub2pdf was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

eshell – start the interactive shell in user context

SYNOPSIS

edrc/pbin/eshell [**-h**]

eshell [*host*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **eshell** command starts **shell**.

This command is located in the **edrc/pbin/** directory to enable the user to conveniently start the **shell** command in his own user context.

See **pf_wrapper**(1) for a description of the benefits of the use of the **edrc/pbin/** directory.

OPTIONS

See: **shell**(1).

ENVIRONMENT

See: **shell**(1).

EXIT STATUS

See: **shell**(1).

FILES

See: **shell**(1).

EXAMPLES

See: **shell**(1).

SEE ALSO

edrcintro(1), **pf_wrapper(1)**, **shell(1)**

NOTES

See: **shell(1)**.

BUGS

See: **shell(1)**.

AUTHOR

eshell was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

eterm – start **edrc**, **asup**, **psup**, **shell**, **eshell** etc. in a separate X-terminal window

SYNOPSIS

edrc/(**pbin**|**bin**)/**eterm** [**-h** | **-l** | **-t**]

eterm [*app*]

eterm [**-i** | **-u**] [*app*]

app ::= **edrc** | **asup** | **osup** | **psup** | **sys** | **lotsctl** | **eshell** | **shell** | ...

AVAILABILITY

WA2L/**edrc**

DESCRIPTION

start application **edrc**, **asup**, **osup**, **psup**, **sys**, **lotsctl**, **eshell**, or **shell** (basically any application available in the **pbin/** directory) in a separate X-terminal window.

The **eterm** command also supports to create **Desktop** icons in the **Gnome** desktop environment.

If **eterm** is started when not logged into a X-session, the given *application* is started in the current shell.

This command is (also) located in the **pbin/** directory to enable the user to conveniently start it.

See **pf_wrapper**(1) for a description of the benefits of the use of the **edrc/pbin/** directory.

OPTIONS

-h usage.

-i install/define an application icon in the **Gnome** desktop environment for the given *app*.

-u uninstall/remove an application icon in the **Gnome** desktop environment for the given *app*.

-l list all defined **Desktop** icons.

-t list x-terminal search list.

app application to start in a separate X-terminal.

The applications available are those available in the **edrc/pbin/** directory: **edrc**, **asup**, **osup**, **psup**, **sys**, **lotsctl**, **eshell**, and **shell**, whereas **edrc** is the default.

ENVIRONMENT

\$ETERM_TERMINALS

space separated list of x-terminals to search (and start). The setting in **\$ETERM_TERMINALS** is prepended to the default search list.

Default on Linux: **gnome-terminal xterm uxterm lxterminal xfce4-terminal tilix mate-terminal konsole sakura urxvt dtterm**.

Default on HP-UX and Solaris: **dtterm xterm**.

\$DISPLAY

X display. If this variable is not set, the current terminal is not started in a X-session.

EXIT STATUS

- | | |
|----------|---|
| 0 | no error. |
| 1 | no terminal application found. |
| 2 | the Desktop directory was not found. |
| 4 | usage displayed. |
| 5 | command aborted. |

FILES

\$HOME/.local/share/application/eterm-app.desktop

icon definition file for the **G**nome desktop environment.

lib/icons/eterm.png

the application icon.

EXAMPLES

-

SEE ALSO

edrcintro(1), **pf_wrapper(1)**

NOTES

-

BUGS

The **yakuake** and **tilda** Linux terminal applications are not supported and ignored if defined due to lack of functionality in context of usage in WA2L/edrc.

The **deepin-terminal** and **lilyterm** Linux terminal applications are supported but not searched by default due to some restrictions in context of usage in WA2L/edrc.

AUTHOR

eterm was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

expect – programmed dialogue with interactive programs, Version 5

SYNOPSIS

expect [**-dDinN**] [**-c** *cmds*] [[**-[f]b**]] *cmdfile*] [*args*]

INTRODUCTION

Expect is a program that "talks" to other interactive programs according to a script. Following the script, **Expect** knows what can be expected from a program and what the correct response should be. An interpreted language provides branching and high-level control structures to direct the dialogue. In addition, the user can take control and interact directly when desired, afterward returning control to the script.

Expectk is a mixture of **Expect** and **Tk**. It behaves just like **Expect** and **Tk**'s **wish**. **Expect** can also be used directly in C or C++ (that is, without Tcl). See libexpect(3).

The name "Expect" comes from the idea of *send/expect* sequences popularized by uucp, kermi and other modem control programs. However unlike uucp, **Expect** is generalized so that it can be run as a user-level command with any program and task in mind. **Expect** can actually talk to several programs at the same time.

For example, here are some things **Expect** can do:

- Cause your computer to dial you back, so that you can login without paying for the call.
- Start a game (e.g., rogue) and if the optimal configuration doesn't appear, restart it (again and again) until it does, then hand over control to you.
- Run fsck, and in response to its questions, answer "yes", "no" or give control back to you, based on predetermined criteria.
- Connect to another network or BBS (e.g., MCI Mail, CompuServe) and automatically retrieve your mail so that it appears as if it was originally sent to your local system.
- Carry environment variables, current directory, or any kind of information across rlogin, telnet, tip, su, chgrp, etc.

There are a variety of reasons why the shell cannot perform these tasks. (Try, you'll see.) All are possible with **Expect**.

In general, **Expect** is useful for running any program which requires interaction between the program and the user. All that is necessary is that the interaction can be characterized programmatically. **Expect** can also give the user back control (without halting the program being controlled) if desired. Similarly, the user can return control to the script at any time.

USAGE

Expect reads *cmdfile* for a list of commands to execute. **Expect** may also be invoked implicitly on systems which support the #! notation by marking the script executable, and making the first line in your script:

```
#!/usr/bin/expect -f
```

Of course, the path must accurately describe where **Expect** lives. /usr/bin is just an example.

The **-c** flag prefaces a command to be executed before any in the script. The command should be quoted to prevent being broken up by the shell. This option may be used multiple times. Multiple commands may be executed with a single **-c** by separating them with semicolons. Commands are executed in the order they appear. (When using Expectk, this option is specified as **-command**.)

The **-d** flag enables some diagnostic output, which primarily reports internal activity of commands such as **expect** and **interact**. This flag has the same effect as "exp_internal 1" at the beginning of an Expect script, plus the version of **Expect** is printed. (The **strace** command is useful for tracing statements, and the **trace** command is useful for tracing variable assignments.) (When using Expectk, this option is specified as **-diag**.)

The **-D** flag enables an interactive debugger. An integer value should follow. The debugger will take control before the next Tcl procedure if the value is non-zero or if a ^C is pressed (or a breakpoint is hit, or other appropriate debugger command appears in the script). See the README file or SEE ALSO (below) for more information on the debugger. (When using Expectk, this option is specified as **-Debug**.)

The **-f** flag prefaces a file from which to read commands from. The flag itself is optional as it is only useful when using the **#!** notation (see above), so that other arguments may be supplied on the command line. (When using Expectk, this option is specified as **-file**.)

By default, the command file is read into memory and executed in its entirety. It is occasionally desirable to read files one line at a time. For example, stdin is read this way. In order to force arbitrary files to be handled this way, use the **-b** flag. (When using Expectk, this option is specified as **-buffer**.) Note that stdio-buffering may still take place however this shouldn't cause problems when reading from a fifo or stdin.

If the string "-" is supplied as a filename, standard input is read instead. (Use "-." to read from a file actually named "-".)

The **-i** flag causes **Expect** to interactively prompt for commands instead of reading them from a file. Prompting is terminated via the **exit** command or upon EOF. See **interpreter** (below) for more information. **-i** is assumed if neither a command file nor **-c** is used. (When using Expectk, this option is specified as **-interactive**.)

-- may be used to delimit the end of the options. This is useful if you want to pass an option-like argument to your script without it being interpreted by **Expect**. This can usefully be placed in the **#!** line to prevent any flag-like interpretation by Expect. For example, the following will leave the original arguments (including the script name) in the variable *argv*.

```
#!/usr/bin/expect --
```

Note that the usual getopt(3) and execve(2) conventions must be observed when adding arguments to the **#!** line.

The file `$exp_library/expect.rc` is sourced automatically if present, unless the **-N** flag is used. (When using Expectk, this option is specified as **-NORC**.) Immediately after this, the file `~/expect.rc` is sourced automatically, unless the **-n** flag is used. If the environment variable DOTDIR is defined, it is treated as a directory and `.expect.rc` is read from there. (When using Expectk, this option is specified as **-norc**.) This sourcing occurs only after executing any **-c** flags.

-v causes Expect to print its version number and exit. (The corresponding flag in Expectk, which uses long flag names, is **-version**.)

Optional *args* are constructed into a list and stored in the variable named *argv*. *argc* is initialized to the length of *argv*.

argv0 is defined to be the name of the script (or binary if no script is used). For example, the following prints out the name of the script and the first three arguments:

```
send_user "$argv0 [lrange $argv 0 2]\n"
```

COMMANDS

Expect uses *Tcl* (Tool Command Language). Tcl provides control flow (e.g., if, for, break), expression evaluation and several other features such as recursion, procedure definition, etc. Commands used here but not defined (e.g., **set**, **if**, **exec**) are Tcl commands (see tcl(3)). **Expect** supports additional commands, described below. Unless otherwise specified, commands return the empty string.

Commands are listed alphabetically so that they can be quickly located. However, new users may find it easier to start by reading the descriptions of **spawn**, **send**, **expect**, and **interact**, in that order.

Note that the best introduction to the language (both Expect and Tcl) is provided in the book "Exploring

Expect" (see SEE ALSO below). Examples are included in this man page but they are very limited since this man page is meant primarily as reference material.

Note that in the text of this man page, "Expect" with an uppercase "E" refers to the **Expect** program while "expect" with a lower-case "e" refers to the **expect** command within the **Expect** program.)

close [*-slave*] [*-onexec 0|1*] [*-i spawn_id*]

closes the connection to the current process. Most interactive programs will detect EOF on their stdin and exit; thus **close** usually suffices to kill the process as well. The **-i** flag declares the process to close corresponding to the named *spawn_id*.

Both **expect** and **interact** will detect when the current process exits and implicitly do a **close**. But if you kill the process by, say, "exec kill \$pid", you will need to explicitly call **close**.

The **-onexec** flag determines whether the spawn id will be closed in any new spawned processes or if the process is overlayed. To leave a spawn id open, use the value 0. A non-zero integer value will force the spawn closed (the default) in any new processes.

The **-slave** flag closes the slave associated with the spawn id. (See "spawn -pty".) When the connection is closed, the slave is automatically closed as well if still open.

No matter whether the connection is closed implicitly or explicitly, you should call **wait** to clear up the corresponding kernel process slot. **close** does not call **wait** since there is no guarantee that closing a process connection will cause it to exit. See **wait** below for more info.

debug [*-now*] *0|1*

controls a Tcl debugger allowing you to step through statements, set breakpoints, etc.

With no arguments, a 1 is returned if the debugger is not running, otherwise a 0 is returned.

With a 1 argument, the debugger is started. With a 0 argument, the debugger is stopped. If a 1 argument is preceded by the **-now** flag, the debugger is started immediately (i.e., in the middle of the **debug** command itself). Otherwise, the debugger is started with the next Tcl statement.

The **debug** command does not change any traps. Compare this to starting Expect with the **-D** flag (see above).

See the README file or SEE ALSO (below) for more information on the debugger.

disconnect

disconnects a forked process from the terminal. It continues running in the background. The process is given its own process group (if possible). Standard I/O is redirected to /dev/null.

The following fragment uses **disconnect** to continue running the script in the background.

```
if {[fork]! = 0} exit
disconnect
...
```

The following script reads a password, and then runs a program every hour that demands a password each time it is run. The script supplies the password so that you only have to type it once. (See the **stty** command which demonstrates how to turn off password echoing.)

```
send_user "password?\n"
expect_user -re "(.*)\n"
for {} 1 {} {
    if {[fork]! = 0} {sleep 3600;continue}
```

```

    disconnect
    spawn priv_prog
    expect Password:
    send "$expect_out(1,string)\r"
    ...
    exit
}

```

An advantage to using **disconnect** over the shell asynchronous process feature (&) is that **Expect** can save the terminal parameters prior to disconnection, and then later apply them to new ptys. With &, **Expect** does not have a chance to read the terminal's parameters since the terminal is already disconnected by the time **Expect** receives control.

exit [*-opts*] [*status*]

causes **Expect** to exit or otherwise prepare to do so.

The **-onexit** flag causes the next argument to be used as an exit handler. Without an argument, the current exit handler is returned.

The **-noexit** flag causes **Expect** to prepare to exit but stop short of actually returning control to the operating system. The user-defined exit handler is run as well as **Expect**'s own internal handlers. No further **Expect** commands should be executed. This is useful if you are running **Expect** with other Tcl extensions. The current interpreter (and main window if in the Tk environment) remain so that other Tcl extensions can clean up. If **Expect**'s **exit** is called again (however this might occur), the handlers are not rerun.

Upon exiting, all connections to spawned processes are closed. Closure will be detected as an EOF by spawned processes. **exit** takes no other actions beyond what the normal `_exit(2)` procedure does. Thus, spawned processes that do not check for EOF may continue to run. (A variety of conditions are important to determining, for example, what signals a spawned process will be sent, but these are system-dependent, typically documented under `exit(3)`.) Spawned processes that continue to run will be inherited by `init`.

status (or 0 if not specified) is returned as the exit status of **Expect**. **exit** is implicitly executed if the end of the script is reached.

exp_continue [*-continue_timer*]

The command **exp_continue** allows **expect** itself to continue executing rather than returning as it normally would. By default **exp_continue** resets the timeout timer. The **-continue_timer** flag prevents timer from being restarted. (See **expect** for more information.)

exp_internal [*-f file*] *value*

causes further commands to send diagnostic information internal to **Expect** to `stderr` if *value* is non-zero. This output is disabled if *value* is 0. The diagnostic information includes every character received, and every attempt made to match the current output against the patterns.

If the optional *file* is supplied, all normal and debugging output is written to that file (regardless of the value of *value*). Any previous diagnostic output file is closed.

The **-info** flag causes **exp_internal** to return a description of the most recent non-info arguments given.

exp_open [*args*] [*-i spawn_id*]

returns a Tcl file identifier that corresponds to the original spawn id. The file identifier can then be used as if it were opened by Tcl's **open** command. (The spawn id should no longer be used. A **wait** should not be executed.

The **-leaveopen** flag leaves the spawn id open for access through Expect commands. A **wait** must be executed on the spawn id.

exp_pid [*-i spawn_id*]

returns the process id corresponding to the currently spawned process. If the **-i** flag is used, the pid returned corresponds to that of the given spawn id.

exp_send

is an alias for **send**.

exp_send_error

is an alias for **send_error**.

exp_send_log

is an alias for **send_log**.

exp_send_tty

is an alias for **send_tty**.

exp_send_user

is an alias for **send_user**.

exp_version [*[-exit] version*]

is useful for assuring that the script is compatible with the current version of Expect.

With no arguments, the current version of **Expect** is returned. This version may then be encoded in your script. If you actually know that you are not using features of recent versions, you can specify an earlier version.

Versions consist of three numbers separated by dots. First is the major number. Scripts written for versions of **Expect** with a different major number will almost certainly not work. **exp_version** returns an error if the major numbers do not match.

Second is the minor number. Scripts written for a version with a greater minor number than the current version may depend upon some new feature and might not run. **exp_version** returns an error if the major numbers match, but the script minor number is greater than that of the running **Expect**.

Third is a number that plays no part in the version comparison. However, it is incremented when the **Expect** software distribution is changed in any way, such as by additional documentation or optimization. It is reset to 0 upon each new minor version.

With the **-exit** flag, **Expect** prints an error and exits if the version is out of date.

expect [*[-opts] pat1 body1*] ... [*[-opts] patn [bodyn]*]

waits until one of the patterns matches the output of a spawned process, a specified time period has passed, or an end-of-file is seen. If the final body is empty, it may be omitted.

Patterns from the most recent **expect_before** command are implicitly used before any other patterns. Patterns from the most recent **expect_after** command are implicitly used after any other patterns.

If the arguments to the entire **expect** statement require more than one line, all the arguments may be "braced" into one so as to avoid terminating each line with a backslash. In this one case, the usual Tcl substitutions will occur despite the braces.

If a pattern is the keyword **eof**, the corresponding body is executed upon end-of-file. If a pattern is the keyword **timeout**, the corresponding body is executed upon timeout. If no timeout keyword is used, an implicit null action is executed upon timeout. The default timeout period is 10 seconds but may be set, for example to 30, by the command "set timeout 30". An infinite timeout may be designated by the value **-1**. If a pattern is the keyword **default**, the corresponding body is executed upon either timeout or end-of-file.

If a pattern matches, then the corresponding body is executed. **expect** returns the result of the body (or the empty string if no pattern matched). In the event that multiple patterns match, the one appearing first is used to select a body.

Each time new output arrives, it is compared to each pattern in the order they are listed. Thus, you may test for absence of a match by making the last pattern something guaranteed to appear, such as a prompt. In situations where there is no prompt, you must use **timeout** (just like you would if you were interacting manually).

Patterns are specified in three ways. By default, patterns are specified as with Tcl's **string match** command. (Such patterns are also similar to C-shell regular expressions usually referred to as "glob" patterns). The **-gl** flag may be used to protect patterns that might otherwise match **expect** flags from doing so. Any pattern beginning with a "-" should be protected this way. (All strings starting with "-" are reserved for future options.)

For example, the following fragment looks for a successful login. (Note that **abort** is presumed to be a procedure defined elsewhere in the script.)

```
expect {
    busy      {puts busy\n ; exp_continue}
    failed    abort
    "invalid password" abort
    timeout   abort
    connected
}
```

Quotes are necessary on the fourth pattern since it contains a space, which would otherwise separate the pattern from the action. Patterns with the same action (such as the 3rd and 4th) require listing the actions again. This can be avoided by using regexp-style patterns (see below). More information on forming glob-style patterns can be found in the Tcl manual.

Regexp-style patterns follow the syntax defined by Tcl's **regexp** (short for "regular expression") command. regexp patterns are introduced with the flag **-re**. The previous example can be rewritten using a regexp as:

```
expect {
    busy      {puts busy\n ; exp_continue}
    -re "failed|invalid password" abort
    timeout   abort
    connected
}
```

Both types of patterns are "unanchored". This means that patterns do not have to match the entire string, but can begin and end the match anywhere in the string (as long as everything else matches). Use **^** to match the beginning of a string, and **\$** to match the end. Note that if you do not wait for the end of a string, your responses can easily end up in the middle of the string as they are echoed from the spawned process. While still producing correct results, the output can look unnatural. Thus, use of **\$** is encouraged if you can exactly describe the characters at the end of a string.

Note that in many editors, the **^** and **\$** match the beginning and end of lines respectively. However, because expect is not line oriented, these characters match the beginning and end of the data (as opposed to lines) currently in the expect matching buffer. (Also, see the note below on "system indigestion.")

The **-ex** flag causes the pattern to be matched as an "exact" string. No interpretation of *****, **^**, etc is made (although the usual Tcl conventions must still be observed). Exact patterns are always unanchored.

The **-nocase** flag causes uppercase characters of the output to compare as if they were lowercase characters. The pattern is not affected.

While reading output, more than 2000 bytes can force earlier bytes to be "forgotten". This may be changed with the function **match_max**. (Note that excessively large values can slow down the pattern matcher.) If *patlist* is **full_buffer**, the corresponding body is executed if *match_max* bytes have been received and no other patterns have matched. Whether or not the **full_buffer** keyword is used, the forgotten characters are written to `expect_out(buffer)`.

If *patlist* is the keyword **null**, and nulls are allowed (via the **remove_nulls** command), the corresponding body is executed if a single ASCII 0 is matched. It is not possible to match 0 bytes via glob or regexp patterns.

Upon matching a pattern (or eof or full_buffer), any matching and previously unmatched output is saved in the variable `expect_out(buffer)`. Up to 9 regexp substring matches are saved in the variables `expect_out(1,string)` through `expect_out(9,string)`. If the **-indices** flag is used before a pattern, the starting and ending indices (in a form suitable for **lrange**) of the 10 strings are stored in the variables `expect_out(X,start)` and `expect_out(X,end)` where X is a digit, corresponds to the substring position in the buffer. 0 refers to strings which matched the entire pattern and is generated for glob patterns as well as regexp patterns. For example, if a process has produced output of "abcdefgh\n", the result of:

```
expect "cd"
```

is as if the following statements had executed:

```
set expect_out(0,string) cd
set expect_out(buffer) abcd
```

and "efgh\n" is left in the output buffer. If a process produced the output "abbbcabkkkka\n", the result of:

```
expect -indices -re "b(b*).*(k+)"
```

is as if the following statements had executed:

```
set expect_out(0,start) 1
set expect_out(0,end) 10
set expect_out(0,string) bbbcabkkkk
set expect_out(1,start) 2
set expect_out(1,end) 3
set expect_out(1,string) bb
set expect_out(2,start) 10
set expect_out(2,end) 10
set expect_out(2,string) k
set expect_out(buffer) abbbcabkkkk
```

and "a\n" is left in the output buffer. The pattern "*" (and **-re ".*"**) will flush the output buffer without reading any more output from the process.

Normally, the matched output is discarded from Expect's internal buffers. This may be prevented by prefixing a pattern with the **-notransfer** flag. This flag is especially useful in experimenting (and can be abbreviated to **-not** for convenience while experimenting).

The spawn id associated with the matching output (or eof or full_buffer) is stored in `expect_out(spawn_id)`.

The **-timeout** flag causes the current expect command to use the following value as a timeout instead of using the value of the timeout variable.

By default, patterns are matched against output from the current process, however the **-i** flag declares the output from the named spawn_id list be matched against any following patterns (up to the next **-i**). The spawn_id list should either be a whitespace separated list of spawn_ids or a variable referring to such a list of spawn_ids.

For example, the following example waits for "connected" from the current process, or "busy", "failed" or "invalid password" from the spawn_id named by \$proc2.

```
expect {
    -i $proc2 busy {puts busy\n ; exp_continue}
    -re "failed|invalid password" abort
    timeout abort
    connected
}
```

The value of the global variable *any_spawn_id* may be used to match patterns to any spawn_ids that are named with all other **-i** flags in the current **expect** command. The spawn_id from a **-i** flag with no associated pattern (i.e., followed immediately by another **-i**) is made available to any other patterns in the same **expect** command associated with *any_spawn_id*.

The **-i** flag may also name a global variable in which case the variable is read for a list of spawn ids. The variable is reread whenever it changes. This provides a way of changing the I/O source while the command is in execution. Spawn ids provided this way are called "indirect" spawn ids.

Actions such as **break** and **continue** cause control structures (i.e., **for**, **proc**) to behave in the usual way. The command **exp_continue** allows **expect** itself to continue executing rather than returning as it normally would.

This is useful for avoiding explicit loops or repeated expect statements. The following example is part of a fragment to automate rlogin. The **exp_continue** avoids having to write a second **expect** statement (to look for the prompt again) if the rlogin prompts for a password.

```
expect {
    Password: {
        stty -echo
        send_user "password (for $user) on $host: "
        expect_user -re "(.*)\n"
        send_user "\n"
        send "$expect_out(1,string)\r"
        stty echo
        exp_continue
    } incorrect {
        send_user "invalid password or account\n"
        exit
    } timeout {
        send_user "connection to $host timed out\n"
        exit
    } eof {
        send_user \
            "connection to host failed: $expect_out(buffer)"
        exit
    } -re $prompt
```

```
}
```

For example, the following fragment might help a user guide an interaction that is already totally automated. In this case, the terminal is put into raw mode. If the user presses "+", a variable is incremented. If "p" is pressed, several returns are sent to the process, perhaps to poke it in some way, and "i" lets the user interact with the process, effectively stealing away control from the script. In each case, the **exp_continue** allows the current **expect** to continue pattern matching after executing the current action.

```
stty raw -echo
expect_after {
    -i $user_spawn_id
    "p" {send "\r\r\r"; exp_continue}
    "+" {incr foo; exp_continue}
    "i" {interact; exp_continue}
    "quit" exit
}
```

By default, **exp_continue** resets the timeout timer. The timer is not restarted, if **exp_continue** is called with the **-continue_timer** flag.

expect_after [*expect_args*]

works identically to the **expect_before** except that if patterns from both **expect** and **expect_after** can match, the **expect** pattern is used. See the **expect_before** command for more information.

expect_background [*expect_args*]

takes the same arguments as **expect**, however it returns immediately. Patterns are tested whenever new input arrives. The pattern **timeout** and **default** are meaningless to **expect_background** and are silently discarded. Otherwise, the **expect_background** command uses **expect_before** and **expect_after** patterns just like **expect** does.

When **expect_background** actions are being evaluated, background processing for the same spawn id is blocked. Background processing is unblocked when the action completes. While background processing is blocked, it is possible to do a (foreground) **expect** on the same spawn id.

It is not possible to execute an **expect** while an **expect_background** is unblocked. **expect_background** for a particular spawn id is deleted by declaring a new **expect_background** with the same spawn id. Declaring **expect_background** with no pattern removes the given spawn id from the ability to match patterns in the background.

expect_before [*expect_args*]

takes the same arguments as **expect**, however it returns immediately. Pattern-action pairs from the most recent **expect_before** with the same spawn id are implicitly added to any following **expect** commands. If a pattern matches, it is treated as if it had been specified in the **expect** command itself, and the associated body is executed in the context of the **expect** command. If patterns from both **expect_before** and **expect** can match, the **expect_before** pattern is used.

If no pattern is specified, the spawn id is not checked for any patterns.

Unless overridden by a **-i** flag, **expect_before** patterns match against the spawn id defined at the time that the **expect_before** command was executed (not when its pattern is matched).

The **-info** flag causes **expect_before** to return the current specifications of what patterns it will match. By default, it reports on the current spawn id. An optional spawn id specification may be given for information on that spawn id. For example

```
expect_before -info -i $proc
```

At most one spawn id specification may be given. The flag `-indirect` suppresses direct spawn ids that come only from indirect specifications.

Instead of a spawn id specification, the flag `"-all"` will cause `"-info"` to report on all spawn ids.

The output of the `-info` flag can be reused as the argument to `expect_before`.

expect_tty [*expect_args*]

is like **expect** but it reads characters from `/dev/tty` (i.e. keystrokes from the user). By default, reading is performed in cooked mode. Thus, lines must end with a return in order for **expect** to see them. This may be changed via **stty** (see the **stty** command below).

expect_user [*expect_args*]

is like **expect** but it reads characters from `stdin` (i.e. keystrokes from the user). By default, reading is performed in cooked mode. Thus, lines must end with a return in order for **expect** to see them. This may be changed via **stty** (see the **stty** command below).

fork creates a new process. The new process is an exact copy of the current **Expect** process. On success, **fork** returns 0 to the new (child) process and returns the process ID of the child process to the parent process. On failure (invariably due to lack of resources, e.g., swap space, memory), **fork** returns -1 to the parent process, and no child process is created.

Forked processes exit via the **exit** command, just like the original process. Forked processes are allowed to write to the log files. If you do not disable debugging or logging in most of the processes, the result can be confusing.

Some pty implementations may be confused by multiple readers and writers, even momentarily. Thus, it is safest to **fork** before spawning processes.

interact [*string1 body1*] ... [*stringn bodyn*]

gives control of the current process to the user, so that keystrokes are sent to the current process, and the `stdout` and `stderr` of the current process are returned.

String-body pairs may be specified as arguments, in which case the body is executed when the corresponding string is entered. (By default, the string is not sent to the current process.) The **interpreter** command is assumed, if the final body is missing.

If the arguments to the entire **interact** statement require more than one line, all the arguments may be "braced" into one so as to avoid terminating each line with a backslash. In this one case, the usual Tcl substitutions will occur despite the braces.

For example, the following command runs **interact** with the following string-body pairs defined: When `^Z` is pressed, **Expect** is suspended. (The `-reset` flag restores the terminal modes.) When `^A` is pressed, the user sees "you typed a control-A" and the process is sent a `^A`. When `$` is pressed, the user sees the date. When `^C` is pressed, **Expect** exits. If "foo" is entered, the user sees "bar". When `~` is pressed, the **Expect** interpreter runs interactively.

```
set CTRLZ \032
interact {
    -reset $CTRLZ {exec kill -STOP [pid]}
    \001 {send_user "you typed a control-A\n";
        send "\001"
    }
    $ {send_user "The date is [clock format [clock seconds]]."}
    \003 exit
    foo {send_user "bar"}
    ~
}
```

In string-body pairs, strings are matched in the order they are listed as arguments. Strings that partially match are not sent to the current process in anticipation of the remainder coming. If characters are then entered such that there can no longer possibly be a match, only the part of the string will be sent to the process that cannot possibly begin another match. Thus, strings that are substrings of partial matches can match later, if the original strings that was attempting to be match ultimately fails.

By default, string matching is exact with no wild cards. (In contrast, the **expect** command uses glob-style patterns by default.) The **-ex** flag may be used to protect patterns that might otherwise match **interact** flags from doing so. Any pattern beginning with a "-" should be protected this way. (All strings starting with "-" are reserved for future options.)

The **-re** flag forces the string to be interpreted as a regexp-style pattern. In this case, matching substrings are stored in the variable *interact_out* similarly to the way **expect** stores its output in the variable **expect_out**. The **-indices** flag is similarly supported.

The pattern **eof** introduces an action that is executed upon end-of-file. A separate **eof** pattern may also follow the **-output** flag in which case it is matched if an eof is detected while writing output. The default **eof** action is "return", so that **interact** simply returns upon any EOF.

The pattern **timeout** introduces a timeout (in seconds) and action that is executed after no characters have been read for a given time. The **timeout** pattern applies to the most recently specified process. There is no default timeout. The special variable "timeout" (used by the **expect** command) has no affect on this timeout.

For example, the following statement could be used to autologout users who have not typed anything for an hour but who still get frequent system messages:

```
interact -input $user_spawn_id timeout 3600 return -output \
    $spawn_id
```

If the pattern is the keyword **null**, and nulls are allowed (via the **remove_nulls** command), the corresponding body is executed if a single ASCII 0 is matched. It is not possible to match 0 bytes via glob or regexp patterns.

Prefacing a pattern with the flag **-iwrite** causes the variable *interact_out(spawn_id)* to be set to the *spawn_id* which matched the pattern (or eof).

Actions such as **break** and **continue** cause control structures (i.e., **for**, **proc**) to behave in the usual way. However **return** causes **interact** to return to its caller, while **inter_return** causes **interact** to cause a return in its caller. For example, if "proc foo" called **interact** which then executed the action **inter_return**, **proc foo** would return. (This means that if **interact** calls **interpreter** interactively typing **return** will cause the **interact** to continue, while **inter_return** will cause the **interact** to return to its caller.)

During **interact**, raw mode is used so that all characters may be passed to the current process. If the current process does not catch job control signals, it will stop if sent a stop signal (by default ^Z). To restart it, send a continue signal (such as by "kill -CONT <pid>"). If you really want to send a SIGSTOP to such a process (by ^Z), consider spawning **cs** first and then running your program. On the other hand, if you want to send a SIGSTOP to **Expect** itself, first call **interpreter** (perhaps by using an escape character), and then press ^Z.

String-body pairs can be used as a shorthand for avoiding having to enter the interpreter and execute commands interactively. The previous terminal mode is used while the body of a string-body pair is being executed.

For speed, actions execute in raw mode by default. The **-reset** flag resets the terminal to the mode it had before **interact** was executed (invariably, cooked mode). Note that characters entered when the mode is being switched may be lost (an unfortunate feature of the terminal driver on some systems). The only reason to use **-reset** is if your action depends on running in cooked mode.

The **-echo** flag sends characters that match the following pattern back to the process that generated them as each character is read. This may be useful when the user needs to see feedback from partially typed patterns.

If a pattern is being echoed but eventually fails to match, the characters are sent to the spawned process. If the spawned process then echoes them, the user will see the characters twice. **-echo** is probably only appropriate in situations where the user is unlikely to not complete the pattern. For example, the following excerpt is from `rftp`, the recursive-ftp script, where the user is prompted to enter `~g`, `~p`, or `~l`, to get, put, or list the current directory recursively. These are so far away from the normal ftp commands, that the user is unlikely to type `~` followed by anything else, except mistakenly, in which case, they'll probably just ignore the result anyway.

```
interact {
    -echo ~g {getcurdirectory 1}
    -echo ~l {getcurdirectory 0}
    -echo ~p {putcurdirectory}
}
```

The **-nobuffer** flag sends characters that match the following pattern on to the output process as characters are read.

This is useful when you wish to let a program echo back the pattern. For example, the following might be used to monitor where a person is dialing (a Hayes-style modem). Each time "atd" is seen the script logs the rest of the line.

```
proc lognumber { } {
    interact -nobuffer -re "(.*)\r" return
    puts $log "[clock format [clock seconds]]: dialed $interact_out(1,string)"
}

interact -nobuffer "atd" lognumber
```

During **interact**, previous use of **log_user** is ignored. In particular, **interact** will force its output to be logged (sent to the standard output) since it is presumed the user doesn't wish to interact blindly.

The **-o** flag causes any following key-body pairs to be applied to the output of the current process. This can be useful, for example, when dealing with hosts that send unwanted characters during a telnet session.

By default, **interact** expects the user to be writing stdin and reading stdout of the **Expect** process itself. The **-u** flag (for "user") makes **interact** look for the user as the process named by its argument (which must be a spawned id).

This allows two unrelated processes to be joined together without using an explicit loop. To aid in debugging, Expect diagnostics always go to stderr (or stdout for certain logging and debugging information). For the same reason, the **interpreter** command will read interactively from stdin.

For example, the following fragment creates a login process. Then it dials the user (not shown), and finally connects the two together. Of course, any process may be substituted for login. A shell, for example, would allow the user to work without supplying an account and password.

```
spawn login
set login $spawn_id
```

```
spawn tip modem
# dial back out to user
# connect user to login
interact -u $login
```

To send output to multiple processes, list each spawn id list prefaced by a **-output** flag. Input for a group of output spawn ids may be determined by a spawn id list prefaced by a **-input** flag. (Both **-input** and **-output** may take lists in the same form as the **-i** flag in the **expect** command, except that any_spawn_id is not meaningful in **interact**.) All following flags and strings (or patterns) apply to this input until another **-input** flag appears. If no **-input** appears, **-output** implies "-input \$user_spawn_id -output". (Similarly, with patterns that do not have **-input**.) If one **-input** is specified, it overrides \$user_spawn_id. If a second **-input** is specified, it overrides \$spawn_id. Additional **-input** flags may be specified.

The two implied input processes default to having their outputs specified as \$spawn_id and \$user_spawn_id (in reverse). If a **-input** flag appears with no **-output** flag, characters from that process are discarded.

The **-i** flag introduces a replacement for the current spawn_id when no other **-input** or **-output** flags are used. A **-i** flag implies a **-o** flag.

It is possible to change the processes that are being interacted with by using indirect spawn ids. (Indirect spawn ids are described in the section on the **expect** command.) Indirect spawn ids may be specified with the **-i**, **-u**, **-input**, or **-output** flags.

interpreter [args]

causes the user to be interactively prompted for **Expect** and Tcl commands. The result of each command is printed.

Actions such as **break** and **continue** cause control structures (i.e., **for**, **proc**) to behave in the usual way. However **return** causes interpreter to return to its caller, while **inter_return** causes **interpreter** to cause a return in its caller. For example, if "proc foo" called **interpreter** which then executed the action **inter_return**, **proc foo** would return. Any other command causes **interpreter** to continue prompting for new commands.

By default, the prompt contains two integers. The first integer describes the depth of the evaluation stack (i.e., how many times Tcl_Eval has been called). The second integer is the Tcl history identifier. The prompt can be set by defining a procedure called "prompt1" whose return value becomes the next prompt. If a statement has open quotes, parens, braces, or brackets, a secondary prompt (by default "+> ") is issued upon newline. The secondary prompt may be set by defining a procedure called "prompt2".

During **interpreter**, cooked mode is used, even if the its caller was using raw mode.

If stdin is closed, **interpreter** will return unless the **-eof** flag is used, in which case the subsequent argument is invoked.

log_file [args] [[-a] file]

If a filename is provided, **log_file** will record a transcript of the session (beginning at that point) in the file. **log_file** will stop recording if no argument is given. Any previous log file is closed.

Instead of a filename, a Tcl file identifier may be provided by using the **-open** or **-leaveopen** flags. This is similar to the **spawn** command. (See **spawn** for more info.)

The **-a** flag forces output to be logged that was suppressed by the **log_user** command.

By default, the **log_file** command *appends* to old files rather than truncating them, for the convenience of being able to turn logging off and on multiple times in one session. To truncate files, use

the **-noappend** flag.

The **-info** flag causes `log_file` to return a description of the most recent non-info arguments given.

log_user *-info|0|1*

By default, the send/expect dialogue is logged to stdout (and a logfile if open). The logging to stdout is disabled by the command "log_user 0" and reenabled by "log_user 1". Logging to the logfile is unchanged.

The **-info** flag causes `log_user` to return a description of the most recent non-info arguments given.

match_max *[-d] [-i spawn_id] [size]*

defines the size of the buffer (in bytes) used internally by **expect**. With no *size* argument, the current size is returned.

With the **-d** flag, the default size is set. (The initial default is 2000.) With the **-i** flag, the size is set for the named spawn id, otherwise it is set for the current process.

overlay *[-# spawn_id] [-# spawn_id] [...] program [args]*

executes *program args* in place of the current **Expect** program, which terminates. A bare hyphen argument forces a hyphen in front of the command name as if it was a login shell. All spawn_ids are closed except for those named as arguments. These are mapped onto the named file identifiers.

Spawn_ids are mapped to file identifiers for the new program to inherit. For example, the following line runs chess and allows it to be controlled by the current process – say, a chess master.

```
overlay -0 $spawn_id -1 $spawn_id -2 $spawn_id chess
```

This is more efficient than "interact -u", however, it sacrifices the ability to do programmed interaction since the **Expect** process is no longer in control.

Note that no controlling terminal is provided. Thus, if you disconnect or remap standard input, programs that do job control (shells, login, etc) will not function properly.

parity *[-d] [-i spawn_id] [value]*

defines whether parity should be retained or stripped from the output of spawned processes. If *value* is zero, parity is stripped, otherwise it is not stripped. With no *value* argument, the current value is returned.

With the **-d** flag, the default parity value is set. (The initial default is 1, i.e., parity is not stripped.) With the **-i** flag, the parity value is set for the named spawn id, otherwise it is set for the current process.

remove_nulls *[-d] [-i spawn_id] [value]*

defines whether nulls are retained or removed from the output of spawned processes before pattern matching or storing in the variable *expect_out* or *interact_out*. If *value* is 1, nulls are removed. If *value* is 0, nulls are not removed. With no *value* argument, the current value is returned.

With the **-d** flag, the default value is set. (The initial default is 1, i.e., nulls are removed.) With the **-i** flag, the value is set for the named spawn id, otherwise it is set for the current process.

Whether or not nulls are removed, **Expect** will record null bytes to the log and stdout.

send *[-flags] string*

Sends *string* to the current process. For example, the command

```
send "hello world\r"
```

sends the characters, h e l l o <blank> w o r l d <return> to the current process. (Tcl includes a printf-like command (called **format**) which can build arbitrarily complex strings.)

Characters are sent immediately although programs with line-buffered input will not read the characters until a return character is sent. A return character is denoted "\r".

The **--** flag forces the next argument to be interpreted as a string rather than a flag. Any string can be preceded by "--" whether or not it actually looks like a flag. This provides a reliable mechanism to specify variable strings without being tripped up by those that accidentally look like flags. (All strings starting with "-" are reserved for future options.)

The **-i** flag declares that the string be sent to the named `spawn_id`. If the `spawn_id` is `user_spawn_id`, and the terminal is in raw mode, newlines in the string are translated to return-newline sequences so that they appear as if the terminal was in cooked mode. The **-raw** flag disables this translation.

The **-null** flag sends null characters (0 bytes). By default, one null is sent. An integer may follow the **-null** to indicate how many nulls to send.

The **-break** flag generates a break condition. This only makes sense if the spawn id refers to a tty device opened via "spawn -open". If you have spawned a process such as `tip`, you should use `tip's` convention for generating a break.

The **-s** flag forces output to be sent "slowly", thus avoid the common situation where a computer outtypes an input buffer that was designed for a human who would never outtype the same buffer. This output is controlled by the value of the variable "send_slow" which takes a two element list. The first element is an integer that describes the number of bytes to send atomically. The second element is a real number that describes the number of seconds by which the atomic sends must be separated. For example, "set send_slow {10 .001}" would force "send -s" to send strings with 1 millisecond in between each 10 characters sent.

The **-h** flag forces output to be sent (somewhat) like a human actually typing. Human-like delays appear between the characters. (The algorithm is based upon a Weibull distribution, with modifications to suit this particular application.) This output is controlled by the value of the variable "send_human" which takes a five element list. The first two elements are average interarrival time of characters in seconds. The first is used by default. The second is used at word endings, to simulate the subtle pauses that occasionally occur at such transitions. The third parameter is a measure of variability where .1 is quite variable, 1 is reasonably variable, and 10 is quite invariable. The extremes are 0 to infinity. The last two parameters are, respectively, a minimum and maximum interarrival time. The minimum and maximum are used last and "clip" the final time. The ultimate average can be quite different from the given average if the minimum and maximum clip enough values.

As an example, the following command emulates a fast and consistent typist:

```
set send_human {.1 .3 1 .05 2}
send -h "I'm hungry. Let's do lunch."
```

while the following might be more suitable after a hangover:

```
set send_human {.4 .4 .2 .5 100}
send -h "Goodd party lash night!"
```

Note that errors are not simulated, although you can set up error correction situations yourself by embedding mistakes and corrections in a send argument.

The flags for sending null characters, for sending breaks, for forcing slow output and for human-

style output are mutually exclusive. Only the one specified last will be used. Furthermore, no *string* argument can be specified with the flags for sending null characters or breaks.

It is a good idea to precede the first **send** to a process by an **expect**. **expect** will wait for the process to start, while **send** cannot. In particular, if the first **send** completes before the process starts running, you run the risk of having your data ignored. In situations where interactive programs offer no initial prompt, you can precede **send** by a delay as in:

```
# To avoid giving hackers hints on how to break in,
# this system does not prompt for an external password.
# Wait for 5 seconds for exec to complete
spawn telnet very.secure.gov
sleep 5
send password\r
```

exp_send is an alias for **send**. If you are using Expecttk or some other variant of Expect in the Tk environment, **send** is defined by Tk for an entirely different purpose. **exp_send** is provided for compatibility between environments. Similar aliases are provided for other Expect's other send commands.

send_error [*-flags*] *string*

is like **send**, except that the output is sent to stderr rather than the current process.

send_log [*--*] *string*

is like **send**, except that the string is only sent to the log file (see **log_file**.) The arguments are ignored if no log file is open.

send_tty [*-flags*] *string*

is like **send**, except that the output is sent to /dev/tty rather than the current process.

send_user [*-flags*] *string*

is like **send**, except that the output is sent to stdout rather than the current process.

sleep *seconds*

causes the script to sleep for the given number of seconds. Seconds may be a decimal number. Interrupts (and Tk events if you are using Expecttk) are processed while Expect sleeps.

spawn [*args*] *program* [*args*]

creates a new process running *program args*. Its stdin, stdout and stderr are connected to Expect, so that they may be read and written by other **Expect** commands. The connection is broken by **close** or if the process itself closes any of the file identifiers.

When a process is started by **spawn**, the variable *spawn_id* is set to a descriptor referring to that process. The process described by *spawn_id* is considered the *current process*. *spawn_id* may be read or written, in effect providing job control.

user_spawn_id is a global variable containing a descriptor which refers to the user. For example, when *spawn_id* is set to this value, **expect** behaves like **expect_user**.

error_spawn_id is a global variable containing a descriptor which refers to the standard error. For example, when *spawn_id* is set to this value, **send** behaves like **send_error**.

tty_spawn_id is a global variable containing a descriptor which refers to /dev/tty. If /dev/tty does not exist (such as in a cron, at, or batch script), then *tty_spawn_id* is not defined. This may be tested as:

```
if {[info vars tty_spawn_id]} {
    # /dev/tty exists
} else {
    # /dev/tty doesn't exist
```

```

    # probably in cron, batch, or at script
}

```

spawn returns the UNIX process id. If no process is spawned, 0 is returned. The variable *spawn_out(slave,name)* is set to the name of the pty slave device.

By default, **spawn** echoes the command name and arguments. The **-noecho** flag stops **spawn** from doing this.

The **-console** flag causes console output to be redirected to the spawned process. This is not supported on all systems.

Internally, **spawn** uses a pty, initialized the same way as the user's tty. This is further initialized so that all settings are "sane" (according to `stty(1)`). If the variable *stty_init* is defined, it is interpreted in the style of `stty` arguments as further configuration. For example, "set *stty_init* raw" will cause further spawned processes's terminals to start in raw mode. **-notttycopy** skips the initialization based on the user's tty. **-nottyinit** skips the "sane" initialization.

Normally, **spawn** takes little time to execute. If you notice **spawn** taking a significant amount of time, it is probably encountering ptys that are wedged. A number of tests are run on ptys to avoid entanglements with errant processes. (These take 10 seconds per wedged pty.) Running Expect with the **-d** option will show if **Expect** is encountering many ptys in odd states. If you cannot kill the processes to which these ptys are attached, your only recourse may be to reboot.

If *program* cannot be spawned successfully because `exec(2)` fails (e.g. when *program* doesn't exist), an error message will be returned by the next **interact** or **expect** command as if *program* had run and produced the error message as output. This behavior is a natural consequence of the implementation of **spawn**. Internally, **spawn** forks, after which the spawned process has no way to communicate with the original **Expect** process except by communication via the *spawn_id*.

The **-open** flag causes the next argument to be interpreted as a Tcl file identifier (i.e., returned by **open**.) The *spawn_id* can then be used as if it were a spawned process. (The file identifier should no longer be used.) This lets you treat raw devices, files, and pipelines as spawned processes without using a pty. 0 is returned to indicate there is no associated process. When the connection to the spawned process is closed, so is the Tcl file identifier. The **-leaveopen** flag is similar to **-open** except that **-leaveopen** causes the file identifier to be left open even after the *spawn_id* is closed.

The **-pty** flag causes a pty to be opened but no process spawned. 0 is returned to indicate there is no associated process. *Spawn_id* is set as usual.

The variable *spawn_out(slave,fd)* is set to a file identifier corresponding to the pty slave. It can be closed using "close -slave".

The **-ignore** flag names a signal to be ignored in the spawned process. Otherwise, signals get the default behavior. Signals are named as in the **trap** command, except that each signal requires a separate flag.

strace *level*

causes following statements to be printed before being executed. (Tcl's `trace` command traces variables.) *level* indicates how far down in the call stack to trace. For example, the following command runs **Expect** while tracing the first 4 levels of calls, but none below that.

```
expect -c "strace 4" script.exp
```

The **-info** flag causes **strace** to return a description of the most recent non-info arguments given.

stty *args*

changes terminal modes similarly to the external stty command.

By default, the controlling terminal is accessed. Other terminals can be accessed by appending "</dev/tty..." to the command. (Note that the arguments should not be grouped into a single argument.)

Requests for status return it as the result of the command. If no status is requested and the controlling terminal is accessed, the previous status of the raw and echo attributes are returned in a form which can later be used by the command.

For example, the arguments **raw** or **-cooked** put the terminal into raw mode. The arguments **-raw** or **cooked** put the terminal into cooked mode. The arguments **echo** and **-echo** put the terminal into echo and noecho mode respectively.

The following example illustrates how to temporarily disable echoing. This could be used in otherwise-automatic scripts to avoid embedding passwords in them. (See more discussion on this under EXPECT HINTS below.)

```
stty -echo
send_user "Password: "
expect_user -re "(.*)\n"
set password $expect_out(1,string)
stty echo
```

system *args*

gives *args* to sh(1) as input, just as if it had been typed as a command from a terminal. **Expect** waits until the shell terminates. The return status from sh is handled the same way that **exec** handles its return status.

In contrast to **exec** which redirects stdin and stdout to the script, **system** performs no redirection (other than that indicated by the string itself). Thus, it is possible to use programs which must talk directly to /dev/tty. For the same reason, the results of **system** are not recorded in the log.

timestamp [*args*]

returns a timestamp. With no arguments, the number of seconds since the epoch is returned.

The **-format** flag introduces a string which is returned but with substitutions made according to the POSIX rules for strftime. For example %a is replaced by an abbreviated weekday name (i.e., Sat). Others are:

| | |
|----|---|
| %a | abbreviated weekday name |
| %A | full weekday name |
| %b | abbreviated month name |
| %B | full month name |
| %c | date-time as in: Wed Oct 6 11:45:56 1993 |
| %d | day of the month (01-31) |
| %H | hour (00-23) |
| %I | hour (01-12) |
| %j | day (001-366) |
| %m | month (01-12) |
| %M | minute (00-59) |
| %p | am or pm |
| %S | second (00-61) |
| %u | day (1-7, Monday is first day of week) |
| %U | week (00-53, first Sunday is first day of week one) |
| %V | week (01-53, ISO 8601 style) |

| | |
|----|---|
| %w | day (0-6) |
| %W | week (00-53, first Monday is first day of week one) |
| %x | date-time as in: Wed Oct 6 1993 |
| %X | time as in: 23:59:59 |
| %y | year (00-99) |
| %Y | year as in: 1993 |
| %Z | timezone (or nothing if not determinable) |
| %% | a bare percent sign |

Other % specifications are undefined. Other characters will be passed through untouched. Only the C locale is supported.

The **–seconds** flag introduces a number of seconds since the epoch to be used as a source from which to format. Otherwise, the current time is used.

The **–gmt** flag forces timestamp output to use the GMT timezone. With no flag, the local timezone is used.

trap *[[command] signals]*

causes the given *command* to be executed upon future receipt of any of the given signals. The command is executed in the global scope. If *command* is absent, the signal action is returned. If *command* is the string SIG_IGN, the signals are ignored. If *command* is the string SIG_DFL, the signals are reset to the system default. *signals* is either a single signal or a list of signals. Signals may be specified numerically or symbolically as per signal(3). The "SIG" prefix may be omitted.

With no arguments (or the argument **–number**), **trap** returns the signal number of the trap command currently being executed.

The **–code** flag uses the return code of the command in place of whatever code Tcl was about to return when the command originally started running.

The **–interp** flag causes the command to be evaluated using the interpreter active at the time the command started running rather than when the trap was declared.

The **–name** flag causes the **trap** command to return the signal name of the trap command currently being executed.

The **–max** flag causes the **trap** command to return the largest signal number that can be set.

For example, the command "trap {send_user "Ouch!"} SIGINT" will print "Ouch!" each time the user presses ^C.

By default, SIGINT (which can usually be generated by pressing ^C) and SIGTERM cause Expect to exit. This is due to the following trap, created by default when Expect starts.

```
trap exit {SIGINT SIGTERM}
```

If you use the **–D** flag to start the debugger, SIGINT is redefined to start the interactive debugger. This is due to the following trap:

```
trap {exp_debug 1} SIGINT
```

The debugger trap can be changed by setting the environment variable EXPECT_DEBUG_INIT to a new trap command.

You can, of course, override both of these just by adding trap commands to your script. In particular, if you have your own "trap exit SIGINT", this will override the debugger trap. This is useful if you want to prevent users from getting to the debugger at all.

If you want to define your own trap on SIGINT but still trap to the debugger when it is running, use:

```
if { ![exp_debug] } { trap mystuff SIGINT }
```

Alternatively, you can trap to the debugger using some other signal.

trap will not let you override the action for SIGALRM as this is used internally to **Expect**. The disconnect command sets SIGALRM to SIG_IGN (ignore). You can reenable this as long as you disable it during subsequent spawn commands.

See signal(3) for more info.

wait [*args*]

delays until a spawned process (or the current process if none is named) terminates.

wait normally returns a list of four integers. The first integer is the pid of the process that was waited upon. The second integer is the corresponding spawn id. The third integer is -1 if an operating system error occurred, or 0 otherwise. If the third integer was 0, the fourth integer is the status returned by the spawned process. If the third integer was -1, the fourth integer is the value of errno set by the operating system. The global variable errorCode is also set.

Additional elements may appear at the end of the return value from **wait**. An optional fifth element identifies a class of information. Currently, the only possible value for this element is CHILD-KILLED in which case the next two values are the C-style signal name and a short textual description.

The **-i** flag declares the process to wait corresponding to the named spawn_id (NOT the process id). Inside a SIGCHLD handler, it is possible to wait for any spawned process by using the spawn id -1.

The **-nowait** flag causes the wait to return immediately with the indication of a successful wait. When the process exits (later), it will automatically disappear without the need for an explicit wait.

The **wait** command may also be used wait for a forked process using the arguments "-i -1". Unlike its use with spawned processes, this command can be executed at any time. There is no control over which process is reaped. However, the return value can be checked for the process id.

LIBRARIES

Expect automatically knows about two built-in libraries for Expect scripts. These are defined by the directories named in the variables exp_library and exp_exec_library. Both are meant to contain utility files that can be used by other scripts.

exp_library contains architecture-independent files. exp_exec_library contains architecture-dependent files. Depending on your system, both directories may be totally empty. The existence of the file \$exp_exec_library/cat-buffers describes whether your /bin/cat buffers by default.

PRETTY-PRINTING

A vgrind definition is available for pretty-printing **Expect** scripts. Assuming the vgrind definition supplied with the **Expect** distribution is correctly installed, you can use it as:

```
vgrind -lexpect file
```

EXAMPLES

It may not be apparent how to put everything together that the man page describes. I encourage you to read and try out the examples in the example directory of the **Expect** distribution. Some of them are real programs. Others are simply illustrative of certain techniques, and of course, a couple are just quick hacks. The `INSTALL` file has a quick overview of these programs.

The **Expect** papers (see `SEE ALSO`) are also useful. While some papers use syntax corresponding to earlier versions of Expect, the accompanying rationales are still valid and go into a lot more detail than this man page.

CAVEATS

Extensions may collide with Expect's command names. For example, **send** is defined by Tk for an entirely different purpose. For this reason, most of the **Expect** commands are also available as `"exp_XXXX"`. Commands and variables beginning with `"exp"`, `"inter"`, `"spawn"`, and `"timeout"` do not have aliases. Use the extended command names if you need this compatibility between environments.

Expect takes a rather liberal view of scoping. In particular, variables read by commands specific to the **Expect** program will be sought first from the local scope, and if not found, in the global scope. For example, this obviates the need to place `"global timeout"` in every procedure you write that uses **expect**. On the other hand, variables written are always in the local scope (unless a `"global"` command has been issued). The most common problem this causes is when `spawn` is executed in a procedure. Outside the procedure, `spawn_id` no longer exists, so the spawned process is no longer accessible simply because of scoping. Add a `"global spawn_id"` to such a procedure.

If you cannot enable the multispawning capability (i.e., your system supports neither `select` (BSD `*.*`), `poll` (SVR<2>), nor something equivalent), **Expect** will only be able to control a single process at a time. In this case, do not attempt to set `spawn_id`, nor should you execute processes via `exec` while a spawned process is running. Furthermore, you will not be able to **expect** from multiple processes (including the user as one) at the same time.

Terminal parameters can have a big effect on scripts. For example, if a script is written to look for echoing, it will misbehave if echoing is turned off. For this reason, Expect forces sane terminal parameters by default. Unfortunately, this can make things unpleasant for other programs. As an example, the emacs shell wants to change the "usual" mappings: newlines get mapped to newlines instead of carriage-return newlines, and echoing is disabled. This allows one to use emacs to edit the input line. Unfortunately, Expect cannot possibly guess this.

You can request that Expect not override its default setting of terminal parameters, but you must then be very careful when writing scripts for such environments. In the case of emacs, avoid depending upon things like echoing and end-of-line mappings.

The commands that accepted arguments braced into a single list (the **expect** variants and **interact**) use a heuristic to decide if the list is actually one argument or many. The heuristic can fail only in the case when the list actually does represent a single argument which has multiple embedded `\n`'s with non-whitespace characters between them. This seems sufficiently improbable, however the argument `"-nobrace"` can be used to force a single argument to be handled as a single argument. This could conceivably be used with machine-generated Expect code. Similarly, `-brace` forces a single argument to be handled as multiple patterns/actions.

BUGS

It was really tempting to name the program `"sex"` (for either `"Smart EXec"` or `"Send-EXpect"`), but good sense (or perhaps just Puritanism) prevailed.

On some systems, when a shell is spawned, it complains about not being able to access the `tty` but runs anyway. This means your system has a mechanism for gaining the controlling `tty` that **Expect** doesn't know

about. Please find out what it is, and send this information back to me.

Ultrix 4.1 (at least the latest versions around here) considers timeouts of above 1000000 to be equivalent to 0.

Digital UNIX 4.0A (and probably other versions) refuses to allocate ptys if you define a SIGCHLD handler. See grantpt page for more info.

IRIX 6.0 does not handle pty permissions correctly so that if Expect attempts to allocate a pty previously used by someone else, it fails. Upgrade to IRIX 6.1.

Telnet (verified only under SunOS 4.1.2) hangs if TERM is not set. This is a problem under cron, at and in cgi scripts, which do not define TERM. Thus, you must set it explicitly - to what type is usually irrelevant. It just has to be set to something! The following probably suffices for most cases.

```
set env(TERM) vt100
```

Tip (verified only under BSDI BSD/OS 3.1 i386) hangs if SHELL and HOME are not set. This is a problem under cron, at and in cgi scripts, which do not define these environment variables. Thus, you must set them explicitly - to what type is usually irrelevant. It just has to be set to something! The following probably suffices for most cases.

```
set env(SHELL) /bin/sh
set env(HOME) /usr/bin
```

Some implementations of ptys are designed so that the kernel throws away any unread output after 10 to 15 seconds (actual number is implementation-dependent) after the process has closed the file descriptor. Thus **Expect** programs such as

```
spawn date
sleep 20
expect
```

will fail. To avoid this, invoke non-interactive programs with **exec** rather than **spawn**. While such situations are conceivable, in practice I have never encountered a situation in which the final output of a truly interactive program would be lost due to this behavior.

On the other hand, Cray UNICOS ptys throw away any unread output immediately after the process has closed the file descriptor. I have reported this to Cray and they are working on a fix.

Sometimes a delay is required between a prompt and a response, such as when a tty interface is changing UART settings or matching baud rates by looking for start/stop bits. Usually, all this is require is to sleep for a second or two. A more robust technique is to retry until the hardware is ready to receive input. The following example uses both strategies:

```
send "speed 9600\r";
sleep 1
expect {
    timeout {send "\r"; exp_continue}
    $prompt
}
```

trap -code will not work with any command that sits in Tcl's event loop, such as sleep. The problem is that in the event loop, Tcl discards the return codes from async event handlers. A workaround is to set a flag in the trap code. Then check the flag immediately after the command (i.e., sleep).

The expect_background command ignores -timeout arguments and has no concept of timeouts in general.

EXPECT HINTS

There are a couple of things about **Expect** that may be non-intuitive. This section attempts to address some of these things with a couple of suggestions.

A common expect problem is how to recognize shell prompts. Since these are customized differently by differently people and different shells, portably automating rlogin can be difficult without knowing the prompt. A reasonable convention is to have users store a regular expression describing their prompt (in particular, the end of it) in the environment variable EXPECT_PROMPT. Code like the following can be used. If EXPECT_PROMPT doesn't exist, the code still has a good chance of functioning correctly.

```
set prompt "(%|#|\\$) $"      ;# default prompt
catch {set prompt $env(EXPECT_PROMPT)}
```

```
expect -re $prompt
```

I encourage you to write **expect** patterns that include the end of whatever you expect to see. This avoids the possibility of answering a question before seeing the entire thing. In addition, while you may well be able to answer questions before seeing them entirely, if you answer early, your answer may appear echoed back in the middle of the question. In other words, the resulting dialogue will be correct but look scrambled.

Most prompts include a space character at the end. For example, the prompt from ftp is 'f', 't', 'p', '>' and <blank>. To match this prompt, you must account for each of these characters. It is a common mistake not to include the blank. Put the blank in explicitly.

If you use a pattern of the form X*, the * will match all the output received from the end of X to the last thing received. This sounds intuitive but can be somewhat confusing because the phrase "last thing received" can vary depending upon the speed of the computer and the processing of I/O both by the kernel and the device driver.

In particular, humans tend to see program output arriving in huge chunks (atomically) when in reality most programs produce output one line at a time. Assuming this is the case, the * in the pattern of the previous paragraph may only match the end of the current line even though there seems to be more, because at the time of the match that was all the output that had been received.

expect has no way of knowing that further output is coming unless your pattern specifically accounts for it.

Even depending on line-oriented buffering is unwise. Not only do programs rarely make promises about the type of buffering they do, but system indigestion can break output lines up so that lines break at seemingly random places. Thus, if you can express the last few characters of a prompt when writing patterns, it is wise to do so.

If you are waiting for a pattern in the last output of a program and the program emits something else instead, you will not be able to detect that with the **timeout** keyword. The reason is that **expect** will not timeout – instead it will get an **eof** indication. Use that instead. Even better, use both. That way if that line is ever moved around, you won't have to edit the line itself.

Newlines are usually converted to carriage return, linefeed sequences when output by the terminal driver. Thus, if you want a pattern that explicitly matches the two lines, from, say, printf("foo\nbar"), you should use the pattern "foo\r\nbar".

A similar translation occurs when reading from the user, via **expect_user**. In this case, when you press return, it will be translated to a newline. If **Expect** then passes that to a program which sets its terminal to raw mode (like telnet), there is going to be a problem, as the program expects a true return. (Some programs are actually forgiving in that they will automatically translate newlines to returns, but most don't.) Unfortunately, there is no way to find out that a program put its terminal into raw mode.

Rather than manually replacing newlines with returns, the solution is to use the command "stty raw", which will stop the translation. Note, however, that this means that you will no longer get the cooked line-editing features.

interact implicitly sets your terminal to raw mode so this problem will not arise then.

It is often useful to store passwords (or other private information) in **Expect** scripts. This is not recommended since anything that is stored on a computer is susceptible to being accessed by anyone. Thus, interactively prompting for passwords from a script is a smarter idea than embedding them literally. Nonetheless, sometimes such embedding is the only possibility.

Unfortunately, the UNIX file system has no direct way of creating scripts which are executable but unreadable. Systems which support setgid shell scripts may indirectly simulate this as follows:

Create the **Expect** script (that contains the secret data) as usual. Make its permissions be 750 (-rwxr-x---) and owned by a trusted group, i.e., a group which is allowed to read it. If necessary, create a new group for this purpose. Next, create a /bin/sh script with permissions 2751 (-rwxr-s---x) owned by the same group as before.

The result is a script which may be executed (and read) by anyone. When invoked, it runs the **Expect** script.

SEE ALSO

Tcl(3), **libexpect(3)**

"Exploring Expect: A Tcl-Based Toolkit for Automating Interactive Programs" by Don Libes, pp. 602, ISBN 1-56592-090-2, O'Reilly and Associates, 1995.

"expect: Curing Those Uncontrollable Fits of Interactivity" by Don Libes, Proceedings of the Summer 1990 USENIX Conference, Anaheim, California, June 11-15, 1990.

"Using expect to Automate System Administration Tasks" by Don Libes, Proceedings of the 1990 USENIX Large Installation Systems Administration Conference, Colorado Springs, Colorado, October 17-19, 1990.

"Tcl: An Embeddable Command Language" by John Ousterhout, Proceedings of the Winter 1990 USENIX Conference, Washington, D.C., January 22-26, 1990.

"expect: Scripts for Controlling Interactive Programs" by Don Libes, Computing Systems, Vol. 4, No. 2, University of California Press Journals, November 1991.

"Regression Testing and Conformance Testing Interactive Programs", by Don Libes, Proceedings of the Summer 1992 USENIX Conference, pp. 135-144, San Antonio, TX, June 12-15, 1992.

"Kibitz - Connecting Multiple Interactive Programs Together", by Don Libes, Software - Practice & Experience, John Wiley & Sons, West Sussex, England, Vol. 23, No. 5, May, 1993.

"A Debugger for Tcl Applications", by Don Libes, Proceedings of the 1993 Tcl/Tk Workshop, Berkeley, CA, June 10-11, 1993.

AUTHOR

Don Libes, National Institute of Standards and Technology

ACKNOWLEDGMENTS

Thanks to John Ousterhout for Tcl, and Scott Paisley for inspiration. Thanks to Rob Savoye for Expect's autoconfiguration code.

The HISTORY file documents much of the evolution of **expect**. It makes interesting reading and might give you further insight to this software. Thanks to the people mentioned in it who sent me bug fixes and gave other assistance.

Design and implementation of **Expect** was paid for in part by the U.S. government and is therefore in the

public domain. However the author and NIST would like credit if this program and documentation or portions of them are used.

NAME

exiftool – Read and write meta information in files

SYNOPSIS**Reading**

exiftool [*OPTIONS*] [*-TAG...*] [*--TAG...*] *FILE...*

Writing

exiftool [*OPTIONS*] *-TAG[+-<]=[VALUE]...* *FILE...*

Copying

exiftool [*OPTIONS*] **-tagsFromFile** *SRCFILE* [*-SRCTAG[>DSTTAG]...*] *FILE...*

Other

exiftool [**-ver** | **-list**[*w|f|r|wf|g[*NUM*]|d|x*]]

For specific examples, see the **EXAMPLES** sections below.

This documentation is displayed if exiftool is run without an input *FILE* when one is expected.

DESCRIPTION

A command-line interface to Image::ExifTool, used for reading and writing meta information in a variety of file types. *FILE* is one or more source file names, directory names, or – for the standard input. Metadata is read from source files and printed in readable form to the console (or written to output text files with **-w**).

To write or delete metadata, tag values are assigned using *-TAG=[VALUE]*, and/or the **-geotag**, **-csv=** or **-json=** options. To copy or move metadata, the **-tagsFromFile** feature is used. By default the original files are preserved with *_original* appended to their names — be sure to verify that the new files are OK before erasing the originals. Once in write mode, exiftool will ignore any read-specific options.

Note: If *FILE* is a directory name then only supported file types in the directory are processed (in write mode only writable types are processed). However, files may be specified by name, or the **-ext** option may be used to force processing of files with any extension. Hidden files in the directory are also processed. Adding the **-r** option causes subdirectories to be processed recursively, but subdirectories with names beginning with “.” are skipped unless **-r** is used.

Below is a list of file types and meta information formats currently supported by ExifTool (r = read, w = write, c = create):

File Types

| | | | | | | | | | |
|------|-----|--------|-------|-------|-------|-------|-----|---------|-----|
| 3FR | r | DR4 | r/w/c | ITC | r | ODP | r | RIFF | r |
| 3G2 | r/w | DSS | r | J2C | r | ODS | r | RSRC | r |
| 3GP | r/w | DV | r | JNG | r/w | ODT | r | RTF | r |
| A | r | DVB | r/w | JP2 | r/w | OFR | r | RW2 | r/w |
| AA | r | DVR-MS | r | JPEG | r/w | OGG | r | RWL | r/w |
| AAE | r | DYLIB | r | JSON | r | OGV | r | RWZ | r |
| AAX | r/w | EIP | r | K25 | r | OPUS | r | RM | r |
| ACR | r | EPS | r/w | KDC | r | ORF | r/w | SEQ | r |
| AFM | r | EPUB | r | KEY | r | OTF | r | SKETCH | r |
| AI | r/w | ERF | r/w | LA | r | PAC | r | SO | r |
| AIFF | r | EXE | r | LFP | r | PAGES | r | SR2 | r/w |
| APE | r | EXIF | r/w/c | LNK | r | PBM | r/w | SRF | r |
| ARQ | r/w | EXR | r | LRV | r/w | PCD | r | SRW | r/w |
| ARW | r/w | EXV | r/w/c | M2TS | r | PCX | r | SVG | r |
| ASF | r | F4A/V | r/w | M4A/V | r/w | PDB | r | SWF | r |
| AVI | r | FFF | r/w | MAX | r | PDF | r/w | THM | r/w |
| AVIF | r/w | FITS | r | MEF | r/w | PEF | r/w | TIFF | r/w |
| AZW | r | FLA | r | MIE | r/w/c | PFA | r | TORRENT | r |
| BMP | r | FLAC | r | MIFF | r | PFB | r | TTC | r |
| BPG | r | FLIF | r/w | MKA | r | PFM | r | TTF | r |

| | | | | | | | | | |
|-------|-----|------|-------|---------|-----|-------|-----|------|-------|
| BTf | r | FLV | r | MKS | r | PGF | r | TXt | r |
| CHM | r | FPF | r | MKV | r | PGM | r/w | VCF | r |
| COS | r | FPX | r | MNG | r/w | PLIST | r | VRD | r/w/c |
| CR2 | r/w | GIF | r/w | MOBI | r | PICT | r | VSD | r |
| CR3 | r/w | GPR | r/w | MODD | r | PMP | r | WAV | r |
| CRM | r/w | GZ | r | MOI | r | PNG | r/w | WDP | r/w |
| CRW | r/w | HDP | r/w | MOS | r/w | PPM | r/w | WEBP | r |
| CS1 | r/w | HDR | r | MOV | r/w | PPT | r | WEBM | r |
| CSV | r | HEIC | r/w | MP3 | r | PPTX | r | WMA | r |
| DCM | r | HEIF | r/w | MP4 | r/w | PS | r/w | WMV | r |
| DCP | r/w | HTML | r | MPC | r | PSB | r/w | WTV | r |
| DCR | r | ICC | r/w/c | MPG | r | PSD | r/w | WV | r |
| DFONT | r | ICS | r | MPO | r/w | PSP | r | X3F | r/w |
| DIVX | r | IDML | r | MQV | r/w | QTIF | r/w | XCF | r |
| DJvU | r | IIQ | r/w | MRW | r/w | R3D | r | XLS | r |
| DLL | r | IND | r/w | MXF | r | RA | r | XLSX | r |
| DNG | r/w | INSP | r/w | NEF | r/w | RAF | r/w | XMP | r/w/c |
| DOC | r | INSV | r | NRW | r/w | RAM | r | ZIP | r |
| DOCX | r | INX | r | NUMBERS | r | RAR | r | | |
| DPX | r | ISO | r | O | r | RAW | r/w | | |

Meta Information

| | | | | | |
|---------------|-------|---------------|-----|--------------|---|
| EXIF | r/w/c | CIFF | r/w | Ricoh RMETA | r |
| GPS | r/w/c | AFCP | r/w | Picture Info | r |
| IPTC | r/w/c | Kodak Meta | r/w | Adobe APP14 | r |
| XMP | r/w/c | FotoStation | r/w | MPF | r |
| MakerNotes | r/w/c | PhotoMechanic | r/w | Stim | r |
| Photoshop IRB | r/w/c | JPEG 2000 | r | DPX | r |
| ICC Profile | r/w/c | DICOM | r | APE | r |
| MIE | r/w/c | Flash | r | Vorbis | r |
| JFIF | r/w/c | FlashPix | r | SPIFF | r |
| Ducky APP12 | r/w/c | QuickTime | r | DjVu | r |
| PDF | r/w/c | Matroska | r | M2TS | r |
| PNG | r/w/c | MXF | r | PE/COFF | r |
| Canon VRD | r/w/c | PrintIM | r | AVCHD | r |
| Nikon Capture | r/w/c | FLAC | r | ZIP | r |
| GeoTIFF | r/w/c | ID3 | r | (and more) | |

OPTIONS

Case is not significant for any command-line option (including tag and group names), except for single-character options when the corresponding upper-case option exists. Many single-character options have equivalent long-name versions (shown in brackets), and some options have inverses which are invoked with a leading double-dash. Unrecognized options are interpreted as tag names (for this reason, multiple single-character options may NOT be combined into one argument). Contrary to standard practice, options may appear after source file names on the exiftool command line.

Option Overview

Tag operations

| | |
|-----------------------|---------------------------------------|
| -TAG or --TAG | Extract or exclude specified tag |
| -TAG[+-^]=[VALUE] | Write new value for tag |
| -TAG[+-]<=DATFILE | Write tag value from contents of file |
| -TAG[+-]<SRCTAG | Copy tag value (see -tagsFromFile) |
| -tagsFromFile SRCFILE | Copy tag values from file |

| | | |
|-------------------------------------|-----------------------|---|
| -x TAG | (-exclude) | Exclude specified tag |
| Input-output text formatting | | |
| -args | (-argFormat) | Format metadata as exiftool arguments |
| -b | (-binary) | Output metadata in binary format |
| -c FMT | (-coordFormat) | Set format for GPS coordinates |
| -charset [[TYPE=]CHARSET] | | Specify encoding for special characters |
| -csv[[+]=CSVFILE] | | Export/import tags in CSV format |
| -d FMT | (-dateFormat) | Set format for date/time values |
| -D | (-decimal) | Show tag ID numbers in decimal |
| -E,-ex,-ec | (-escape(HTML XML C)) | Escape tag values for HTML, XML or C |
| -f | (-forcePrint) | Force printing of all specified tags |
| -g[NUM...] | (-groupHeadings) | Organize output by tag group |
| -G[NUM...] | (-groupNames) | Print group name for each tag |
| -h | (-htmlFormat) | Use HTML formatting for output |
| -H | (-hex) | Show tag ID numbers in hexadecimal |
| -htmlDump[OFFSET] | | Generate HTML-format binary dump |
| -j[[+]=JSONFILE] (-json) | | Export/import tags in JSON format |
| -l | (-long) | Use long 2-line output format |
| -L | (-latin) | Use Windows Latin1 encoding |
| -lang [LANG] | | Set current language |
| -listItem INDEX | | Extract specific item from a list |
| -n | (--printConv) | No print conversion |
| -p FMTFILE | (-printFormat) | Print output in specified format |
| -php | | Export tags as a PHP Array |
| -s[NUM] | (-short) | Short output format |
| -S | (-veryShort) | Very short output format |
| -sep STR | (-separator) | Set separator string for list items |
| -sort | | Sort output alphabetically |
| -struct | | Enable output of structured information |
| -t | (-tab) | Output in tab-delimited list format |
| -T | (-table) | Output in tabular format |
| -v[NUM] | (-verbose) | Print verbose messages |
| -w[+ !] EXT | (-textOut) | Write (or overwrite!) output text files |
| -W[+ !] FMT | (-tagOut) | Write output text file for each tag |
| -Wext EXT | (-tagOutExt) | Write only specified file types with -W |
| -X | (-xmlFormat) | Use RDF/XML output format |
| Processing control | | |
| -a | (-duplicates) | Allow duplicate tags to be extracted |
| -e | (--composite) | Do not generate composite tags |
| -ee | (-extractEmbedded) | Extract information from embedded files |
| -ext[+] EXT | (-extension) | Process files with specified extension |
| -F[OFFSET] | (-fixBase) | Fix the base for maker notes offsets |
| -fast[NUM] | | Increase speed when extracting metadata |
| -fileOrder[NUM] [-]TAG | | Set file processing order |
| -i DIR | (-ignore) | Ignore specified directory name |
| -if[NUM] EXPR | | Conditionally process files |
| -m | (-ignoreMinorErrors) | Ignore minor errors and warnings |
| -o OUTFILE | (-out) | Set output file or directory name |
| -overwrite_original | | Overwrite original by renaming tmp file |
| -overwrite_original_in_place | | Overwrite original by copying tmp file |
| -P | (-preserve) | Preserve file modification date/time |
| -password PASSWD | | Password for processing protected files |
| -progress[:[TITLE]] | | Show file progress count |

| | | |
|--------------------------|---------------------------|------------------------------------|
| <code>-q</code> | <code>(-quiet)</code> | Quiet processing |
| <code>-r[.]</code> | <code>(-recurse)</code> | Recursively process subdirectories |
| <code>-scanForXMP</code> | | Brute force XMP scan |
| <code>-u</code> | <code>(-unknown)</code> | Extract unknown tags |
| <code>-U</code> | <code>(-unknown2)</code> | Extract unknown binary tags too |
| <code>-wm MODE</code> | <code>(-writeMode)</code> | Set mode for writing/creating tags |
| <code>-z</code> | <code>(-zip)</code> | Read/write compressed information |

Other options

| | | |
|---|-----------------------|---------------------------------------|
| <code>-@ ARGFILE</code> | | Read command-line arguments from file |
| <code>-k</code> | <code>(-pause)</code> | Pause before terminating |
| <code>-list [w f wf g[<i>NUM</i>] d x]</code> | | List various exiftool capabilities |
| <code>-ver</code> | | Print exiftool version number |
| <code>--</code> | | End of options |

Special features

| | | |
|-------------------------------------|--|--------------------------------------|
| <code>-geotag TRKFILE</code> | | Geotag images from specified GPS log |
| <code>-globalTimeShift SHIFT</code> | | Shift all formatted date/time values |
| <code>-use MODULE</code> | | Add features from plug-in module |

Utilities

| | | |
|----------------------------------|--|----------------------------------|
| <code>-delete_original[!]</code> | | Delete "_original" backups |
| <code>-restore_original</code> | | Restore from "_original" backups |

Advanced options

| | | |
|---|--|--|
| <code>-api OPT[[^]=[VAL]]</code> | | Set ExifTool API option |
| <code>-common_args</code> | | Define common arguments |
| <code>-config CFGFILE</code> | | Specify configuration file name |
| <code>-echo[<i>NUM</i>] TEXT</code> | | Echo text to stdout or stderr |
| <code>-efile[<i>NUM</i>] [!] ERRFILE</code> | | Save names of files with errors |
| <code>-execute[<i>NUM</i>]</code> | | Execute multiple commands on one line |
| <code>-srcfile FMT</code> | | Process a different source file |
| <code>-stay_open FLAG</code> | | Keep reading -@ argfile even after EOF |
| <code>-userParam PARAM[[^]=[VAL]]</code> | | Set user parameter (API UserParam opt) |

Option Details

Tag operations

`-TAG`

Extract information for the specified tag (eg. `-CreateDate`). Multiple tags may be specified in a single command. A tag name is the handle by which a piece of information is referenced. See `Image::ExifTool::TagNames` for documentation on available tag names. A tag name may include leading group names separated by colons (eg. `-EXIF:CreateDate`, or `-Doc1:XMP:Creator`), and each group name may be prefixed by a digit to specify family number (eg. `-1IPTC:City`). Use the **-listg** option to list available group names by family.

A special tag name of `All` may be used to indicate all meta information (ie. **-All**). This is particularly useful when a group name is specified to extract all information in a group (but beware that unless the **-a** option is also used, some tags in the group may be suppressed by same-named tags in other groups). The wildcard characters `?` and `*` may be used in a tag name to match any single character and zero or more characters respectively. These may not be used in a group name, with the exception that a group name of `*` (or `All`) may be used to extract all instances of a tag (as if **-a** was used). Note that arguments containing wildcards must be quoted on the command line of most systems to prevent shell globbing.

A `#` may be appended to the tag name to disable the print conversion on a per-tag basis (see the **-n** option). This may also be used when writing or copying tags.

If no tags are specified, all available information is extracted (as if `-All` had been specified).

Note: Descriptions, not tag names, are shown by default when extracting information. Use the `-s` option to see the tag names instead.

`--TAG`

Exclude specified tag from extracted information. Same as the `-x` option. Group names and wildcards are permitted as described above for `-TAG`. Once excluded from the output, a tag may not be re-included by a subsequent option. May also be used following a `-tagsFromFile` option to exclude tags from being copied (when redirecting to another tag, it is the source tag that should be excluded), or to exclude groups from being deleted when deleting all information (eg. `-all=--exif:all` deletes all but EXIF information). But note that this will not exclude individual tags from a group delete (unless a family 2 group is specified, see note 4 below). Instead, individual tags may be recovered using the `-tagsFromFile` option (eg. `-all= -tagsfromfile @ -artist`).

`-TAG[+-^]=[VALUE]`

Write a new value for the specified tag (eg. `-comment=wow`), or delete the tag if no *VALUE* is given (eg. `-comment=`). `+=` and `-=` are used to add or remove existing entries from a list, or to shift date/time values (see `Image::ExifTool::Shift.pl` and note 6 below for more details). `+=` may also be used to increment numerical values (or decrement if *VALUE* is negative), and `-=` may be used to conditionally delete or replace a tag (see “WRITING EXAMPLES” for examples). `^=` is used to write an empty string instead of deleting the tag when no *VALUE* is given, but otherwise it is equivalent to `=`.

TAG may contain one or more leading family 0, 1 or 2 group names, prefixed by optional family numbers, and separated colons. If no group name is specified, the tag is created in the preferred group, and updated in any other location where a same-named tag already exists. The preferred group is the first group in the following list where *TAG* is valid: 1) EXIF, 2) IPTC, 3) XMP.

The wildcards `*` and `?` may be used in tag names to assign the same value to multiple tags. When specified with wildcards, “unsafe” tags are not written. A tag name of `All` is equivalent to `*` (except that it doesn’t require quoting, while arguments with wildcards do on systems with shell globbing), and is often used when deleting all metadata (ie. `-All=`) or an entire group (eg. `-XMP-dc:All=`, see note 4 below). Note that not all groups are deletable, and that the JPEG APP14 “Adobe” group is not removed by default with `-All=` because it may affect the appearance of the image. However, color space information is removed, so the colors may be affected (but this may be avoided by copying back the tags defined by the `ColorSpaceTags` shortcut). Use the `-listd` option for a complete list of deletable groups, and see note 5 below regarding the “APP” groups. Also, within an image some groups may be contained within others, and these groups are removed if the containing group is deleted:

JPEG Image:

- Deleting EXIF or IFD0 also deletes ExifIFD, GlobParamIFD, GPS, IFD1, InteropIFD, MakerNotes, PrintIM and SubIFD.
- Deleting ExifIFD also deletes InteropIFD and MakerNotes.
- Deleting Photoshop also deletes IPTC.

TIFF Image:

- Deleting EXIF only removes ExifIFD which also deletes InteropIFD and MakerNotes.

Notes:

1) **Many tag values may be assigned in a single command.** If two assignments affect the same tag, the latter takes precedence (except for list-type tags, for which both values are written).

2) In general, MakerNotes tags are considered “Permanent”, and may be edited but not created or deleted individually. This avoids many potential problems, including the inevitable compatibility problems with OEM software which may be very inflexible about the information it expects to find in

the maker notes.

3) Changes to PDF files by ExifTool are reversible (by deleting the update with `-PDF-update:all=`) because the original information is never actually deleted from the file. So ExifTool alone may not be used to securely edit metadata in PDF files.

4) Specifying `-GROUP:all=` deletes the entire group as a block only if a single family 0 or 1 group is specified. Otherwise all deletable tags in the specified group(s) are removed individually, and in this case it is possible to exclude individual tags from a mass delete. For example, `-time:all --Exif:Time:All` removes all deletable Time tags except those in the EXIF. This difference also applies if family 2 is specified when deleting all groups. For example, `-2all:all=` deletes tags individually, while `-all:all=` deletes entire blocks.

5) The “APP” group names (“APP0” through “APP15”) are used to delete JPEG application segments which are not associated with another deletable group. For example, specifying `-APP14:All=` will NOT delete the APP14 “Adobe” segment because this is accomplished with `-Adobe:All`.

6) When shifting a value, the shift is applied to the original value of the tag, overriding any other values previously assigned to the tag on the same command line. To shift a date/time value and copy it to another tag in the same operation, use the `-globalTimeShift` option.

Special feature: Integer values may be specified in hexadecimal with a leading 0x, and simple rational values may be specified as fractions.

`-TAG<=DATFILE` or `-TAG<=FMT`

Set the value of a tag from the contents of file *DATFILE*. The file name may also be given by a *FMT* string where %d, %f and %e represent the directory, file name and extension of the original *FILE* (see the `-w` option for more details). Note that quotes are required around this argument to prevent shell redirection since it contains a < symbol. If *DATFILE/FMT* is not provided, the effect is the same as `-TAG=`, and the tag is simply deleted. `+<=` or `-<=` may also be used to add or delete specific list entries, or to shift date/time values.

`-tagsFromFile SRCFILE` or *FMT*

Copy tag values from *SRCFILE* to *FILE*. Tag names on the command line after this option specify the tags to be copied, or excluded from the copy. Wildcards are permitted in these tag names. If no tags are specified, then all possible tags (see note 1 below) from the source file are copied to same-named tags in the preferred location of the output file (the same as specifying `-all`). More than one `-tagsFromFile` option may be used to copy tags from multiple files.

By default, this option will update any existing and writable same-named tags in the output *FILE*, but will create new tags only in their preferred groups. This allows some information to be automatically transferred to the appropriate group when copying between images of different formats. However, if a group name is specified for a tag then the information is written only to this group (unless redirected to another group, see below). If *All* is used as a group name, then the specified tag(s) are written to the same family 1 group they had in the source file (ie. the same specific location, like ExifIFD or XMP-dc). For example, the common operation of copying all writable tags to the same specific locations in the output *FILE* is achieved by adding `-all:all`. A different family may be specified by adding a leading family number to the group name (eg. `-0all:all` preserves the same general location, like EXIF or XMP).

SRCFILE may be the same as *FILE* to move information around within a single file. In this case, @ may be used to represent the source file (ie. `-tagsFromFile @`), permitting this feature to be used for batch processing multiple files. Specified tags are then copied from each file in turn as it is rewritten. For advanced batch use, the source file name may also be specified using a *FMT* string in which %d, %f and %e represent the directory, file name and extension of *FILE*. (eg. the current *FILE* would be represented by %d%f.%e, with the same effect as @). See the `-w` option for *FMT* string examples.

A powerful redirection feature allows a destination tag to be specified for each copied tag. With this feature, information may be written to a tag with a different name or group. This is done using

""-DSTTAG<SRCTAG"" or ""-SRCTAG>DSTTAG"" on the command line after **-tagsFromFile**, and causes the value of *SRCTAG* to be copied from *SRCFILE* and written to *DSTTAG* in *FILE*. Has no effect unless *SRCTAG* exists in *SRCFILE*. Note that this argument must be quoted to prevent shell redirection, and there is no = sign as when assigning new values. Source and/or destination tags may be prefixed by a group name and/or suffixed by #. Wildcards are allowed in both the source and destination tag names. A destination group and/or tag name of All or * writes to the same family 1 group and/or tag name as the source. If no destination group is specified, the information is written to the preferred group. Whitespace around the > or < is ignored. As a convenience, **-tagsFromFile @** is assumed for any redirected tags which are specified without a prior **-tagsFromFile** option. Copied tags may also be added or deleted from a list with arguments of the form ""-SRCTAG+<DSTTAG"" or ""-SRCTAG-<DSTTAG"" (but see Note 5 below).

An extension of the redirection feature allows strings involving tag names to be used on the right hand side of the < symbol with the syntax ""-DSTTAG<STR"", where tag names in *STR* are prefixed with a \$ symbol. See the **-p** option and the "Advanced formatting feature" section for more details about this syntax. Strings starting with a = sign must insert a single space after the < to avoid confusion with the <= operator which sets the tag value from the contents of a file. A single space at the start of the string is removed if it exists, but all other whitespace in the string is preserved. See note 8 below about using the redirection feature with list-type tags, shortcuts or when using wildcards in tag names.

See "COPYING EXAMPLES" for examples using **-tagsFromFile**.

Notes:

- 1) Some tags (generally tags which may affect the appearance of the image) are considered "unsafe" to write, and are only copied if specified explicitly (ie. no wildcards). See the tag name documentation for more details about "unsafe" tags.
- 2) Be aware of the difference between excluding a tag from being copied (**--TAG**), and deleting a tag (**-TAG=**). Excluding a tag prevents it from being copied to the destination image, but deleting will remove a pre-existing tag from the image.
- 3) The maker note information is copied as a block, so it isn't affected like other information by subsequent tag assignments on the command line, and individual makernote tags may not be excluded from a block copy. Also, since the PreviewImage referenced from the maker notes may be rather large, it is not copied, and must be transferred separately if desired.
- 4) The order of operations is to copy all specified tags at the point of the **-tagsFromFile** option in the command line. Any tag assignment to the right of the **-tagsFromFile** option is made after all tags are copied. For example, new tag values are set in the order One, Two, Three then Four with this command:

```
exiftool -One=1 -tagsFromFile s.jpg -Two -Four=4 -Three d.jpg
```

This is significant in the case where an overlap exists between the copied and assigned tags because later operations may override earlier ones.

- 5) The normal behaviour of copied tags differs from that of assigned tags for list-type tags and conditional replacements because each copy operation on a tag overrides any previous operations. While this avoids duplicate list items when copying groups of tags from a file containing redundant information, it also prevents values of different tags from being copied into the same list when this is the intent. So a **-addTagsFromFile** option is provided which allows copying of multiple tags into the same list. eg)

```
exiftool -addtagsfromfile @ '-subject<make' '-subject<model' ...
```

Similarly, **-addTagsFromFile** must be used when conditionally replacing a tag to prevent overriding earlier conditions.

Other than these differences, the **-tagsFromFile** and **-addTagsFromFile** options are equivalent.

6) The **-a** option (allow duplicate tags) is always in effect when copying tags from *SRCFILE*, but the highest priority tag is always copied last so it takes precedence.

7) Structured tags are copied by default when copying tags. See the **-struct** option for details.

8) With the redirection feature, copying a tag directly (ie. `"-DSTTAG<SRCTAG"`) is not the same as interpolating its value inside a string (ie. `"-DSTTAG<$SRCTAG"`) for list-type tags, shortcut tags, tag names containing wildcards, or UserParam variables. When copying directly, the values of each matching source tag are copied individually to the destination tag (as if they were separate assignments). However, when interpolated inside a string, list items and the values of shortcut tags are concatenated (with a separator set by the **-sep** option), and wildcards are not allowed. Also, UserParam variables are available only when interpolated in a string. Another difference is that a minor warning is generated if a tag doesn't exist when interpolating its value in a string (with `$`), but isn't when copying the tag directly.

Finally, the behaviour is different when a destination tag or group of All is used. When copying directly, a destination group and/or tag name of All writes to the same family 1 group and/or tag name as the source. But when interpolated in a string, the identity of the source tags are lost and the value is written to all possible groups/tags. For example, the string form must be used in the following command since the intent is to set the value of all existing date/time tags from `CreateDate`:

```
exiftool "-time:all<$createdate" -wm w FILE
```

-x TAG (-exclude)

Exclude the specified tag. There may be multiple **-x** options. This has the same effect as `---TAG` on the command line. See the `---TAG` documentation above for a complete description.

Input-output text formatting

Note that trailing spaces are removed from extracted values for most output text formats. The exceptions are **-b**, **-csv**, **-j** and **-X**.

-args (-argFormat)

Output information in the form of exiftool arguments, suitable for use with the **-@** option when writing. May be combined with the **-G** option to include group names. This feature may be used to effectively copy tags between images, but allows the metadata to be altered by editing the intermediate file (`out.args` in this example):

```
exiftool -args -G1 --filename --directory src.jpg > out.args
exiftool -@ out.args -sep ", " dst.jpg
```

Note: Be careful when copying information with this technique since it is easy to write tags which are normally considered "unsafe". For instance, the `FileName` and `Directory` tags are excluded in the example above to avoid renaming and moving the destination file. Also note that the second command above will produce warning messages for any tags which are not writable.

As well, the **-sep** option should be used as in the second command above to maintain separate list items when writing metadata back to image files, and the **-struct** option may be used when extracting to preserve structured XMP information.

-b (-binary)

Output requested metadata in binary format without tag names or descriptions. This option is mainly used for extracting embedded images or other binary data, but it may also be useful for some text strings since control characters (such as newlines) are not replaced by `'.'` as they are in the default output. By default, list items are separated by a newline when extracted with the **-b** option, but this may be changed (see the **-sep** option for details). May be combined with **-j**, **-php** or **-X** to extract binary data in JSON, PHP or XML format, but note that "unsafe" tags must be specified explicitly to be extracted as binary in these formats.

-c *FMT* (-coordFormat)

Set the print format for GPS coordinates. *FMT* uses the same syntax as a `printf` format string. The specifiers correspond to degrees, minutes and seconds in that order, but minutes and seconds are optional. For example, the following table gives the output for the same coordinate using various formats:

| FMT | Output |
|---------------------|---|
| ----- | ----- |
| "%d deg %d' %.2f\"" | 54 deg 59' 22.80" (default for reading) |
| "%d %d %.8f" | 54 59 22.80000000 (default for copying) |
| "%d deg %.4f min" | 54 deg 59.3800 min |
| "%.6f degrees" | 54.989667 degrees |

Notes:

- 1) To avoid loss of precision, the default coordinate format is different when copying tags using the **-tagsFromFile** option.
- 2) If the hemisphere is known, a reference direction (N, S, E or W) is appended to each printed coordinate, but adding a + to the format specifier (eg. `%+.6f`) prints a signed coordinate instead.
- 3) This print formatting may be disabled with the **-n** option to extract coordinates as signed decimal degrees.

-charset [*TYPE*=]*CHARSET*

If *TYPE* is `ExifTool` or not specified, this option sets the ExifTool character encoding for output tag values when reading and input values when writing, with a default of UTF8. If no *CHARSET* is given, a list of available character sets is returned. Valid *CHARSET* values are:

| CHARSET | Alias(es) | Description |
|-------------|-----------------|-----------------------------------|
| ----- | ----- | ----- |
| UTF8 | cp65001, UTF-8 | UTF-8 characters (default) |
| Latin | cp1252, Latin1 | Windows Latin1 (West European) |
| Latin2 | cp1250 | Windows Latin2 (Central European) |
| Cyrillic | cp1251, Russian | Windows Cyrillic |
| Greek | cp1253 | Windows Greek |
| Turkish | cp1254 | Windows Turkish |
| Hebrew | cp1255 | Windows Hebrew |
| Arabic | cp1256 | Windows Arabic |
| Baltic | cp1257 | Windows Baltic |
| Vietnam | cp1258 | Windows Vietnamese |
| Thai | cp874 | Windows Thai |
| DOSLatinUS | cp437 | DOS Latin US |
| DOSLatin1 | cp850 | DOS Latin1 |
| DOSCyrillic | cp866 | DOS Cyrillic |
| MacRoman | cp10000, Roman | Macintosh Roman |
| MacLatin2 | cp10029 | Macintosh Latin2 (Central Europe) |
| MacCyrillic | cp10007 | Macintosh Cyrillic |
| MacGreek | cp10006 | Macintosh Greek |
| MacTurkish | cp10081 | Macintosh Turkish |
| MacRomanian | cp10010 | Macintosh Romanian |
| MacIceland | cp10079 | Macintosh Icelandic |
| MacCroatian | cp10082 | Macintosh Croatian |

TYPE may be `FileName` to specify the encoding of file names on the command line (ie. *FILE* arguments). In Windows, this triggers use of wide-character i/o routines, thus providing support for Unicode file names. See the "WINDOWS UNICODE FILE NAMES" section below for details.

Other values of *TYPE* listed below are used to specify the internal encoding of various meta

information formats.

| TYPE | Description | Default |
|-----------|--|----------|
| ----- | ----- | ----- |
| EXIF | Internal encoding of EXIF "ASCII" strings | (none) |
| ID3 | Internal encoding of ID3v1 information | Latin |
| IPTC | Internal IPTC encoding to assume when IPTC:CodedCharacterSet is not defined | Latin |
| Photoshop | Internal encoding of Photoshop IRB strings | Latin |
| QuickTime | Internal encoding of QuickTime strings | MacRoman |
| RIFF | Internal encoding of RIFF strings | 0 |

See <<https://exiftool.org/faq.html#Q10>> for more information about coded character sets, and the Image::ExifTool Options for more details about the **-charset** settings.

-csv[**[+]=CSVFILE**]

Export information in CSV format, or import information if *CSVFILE* is specified. When importing, the CSV file must be in exactly the same format as the exported file. The first row of the *CSVFILE* must be the ExifTool tag names (with optional group names) for each column of the file, and values must be separated by commas. A special "SourceFile" column specifies the files associated with each row of information (and a SourceFile of "*" may be used to define default tags to be imported for all files which are combined with any tags specified for the specific SourceFile processed). The following examples demonstrate basic use of this option:

```
# generate CSV file with common tags from all images in a directory
exiftool -common -csv dir > out.csv

# update metadata for all images in a directory from CSV file
exiftool -csv=a.csv dir
```

Empty values are ignored when importing (unless the **-f** option is used and the API MissingTagValue is set to an empty string, in which case the tag is deleted). Also, FileName and Directory columns are ignored if they exist (ie. ExifTool will not attempt to write these tags with a CSV import). To force a tag to be deleted, use the **-f** option and set the value to "-" in the CSV file (or to the MissingTagValue if this API option was used). Multiple databases may be imported in a single command.

When exporting a CSV file, the **-g** or **-G** option adds group names to the tag headings. If the **-a** option is used to allow duplicate tag names, the duplicate tags are only included in the CSV output if the column headings are unique. Adding the **-G4** option ensures a unique column heading for each tag. The **-b** option may be added to output binary data, encoded in base64 if necessary (indicated by ASCII "base64:" as the first 7 bytes of the value). Values may also be encoded in base64 if the **-charset** option is used and the value contains invalid characters.

When exporting specific tags, the CSV columns are arranged in the same order as the specified tags provided the column headings exactly match the specified tag names, otherwise the columns are sorted in alphabetical order.

When importing from a CSV file, only files specified on the command line are processed. Any extra entries in the CSV file are ignored.

List-type tags are stored as simple strings in a CSV file, but the **-sep** option may be used to split them back into separate items when importing.

Special feature: **-csv+=CSVFILE** may be used to add items to existing lists. This affects only list-type tags. Also applies to the **-j** option.

Note that this option is fundamentally different than all other output format options because it requires information from all input files to be buffered in memory before the output is written. This may result in excessive memory usage when processing a very large number of files with a single

command. Also, it makes this option incompatible with the **-w** option. When processing a large number of files, it is recommended to either use the JSON (**-j**) or XML (**-X**) output format, or use **-p** to generate a fixed-column CSV file instead of using the **-csv** option.

-d FMT (**-dateFormat**)

Set the format for date/time tag values. The *FMT* string may contain formatting codes beginning with a percent character (%) to represent the various components of a date/time value. The specifics of the *FMT* syntax are system dependent — consult the `strftime` man page on your system for details. The default format is equivalent to “%Y:%m:%d %H:%M:%S”. This option has no effect on date-only or time-only tags and ignores timezone information if present. Only one **-d** option may be used per command. Requires POSIX::strftime or Time::Piece for the inversion conversion when writing.

-D (**-decimal**)

Show tag ID number in decimal when extracting information.

-E, -ex, -ec (**-escapeHTML, -escapeXML, -escapeC**)

Escape characters in output tag values for HTML (**-E**), XML (**-ex**) or C (**-ec**). For HTML, all characters with Unicode code points above U+007F are escaped as well as the following 5 characters: & (') " (") > (>) and < (<). For XML, only these 5 characters are escaped. The **-E** option is implied with **-h**, and **-ex** is implied with **-X**. For C, all control characters and the backslash are escaped. The inverse conversion is applied when writing tags.

-f (**-forcePrint**)

Force printing of tags even if their values are not found. This option only applies when specific tags are requested on the command line (ie. not with wildcards or by **-all**). With this option, a dash (-) is printed for the value of any missing tag, but the dash may be changed via the `API MissingTagValue` option. May also be used to add a 'flags' attribute to the **-listx** output, or to allow tags to be deleted when writing with the **-csv=CSVFILE** feature.

-g[*NUM*][:*NUM*...] (**-groupHeadings**)

Organize output by tag group. *NUM* specifies a group family number, and may be 0 (general location), 1 (specific location), 2 (category), 3 (document number), 4 (instance number), 5 (metadata path) or 6 (EXIF/TIFF format). **-g0** is assumed if a family number is not specified, and family numbers may be added wherever **-g** is mentioned in the documentation. Multiple families may be specified by separating them with colons. By default the resulting group name is simplified by removing any leading `Main:` and collapsing adjacent identical group names, but this can be avoided by placing a colon before the first family number (eg. **-g:3:1**). Use the **-listg** option to list group names for a specified family. The `SavePath` and `SaveFormat` API options are automatically enabled if the respective family 5 or 6 group names are requested. See the `API GetGroup` documentation for more information.

-G[*NUM*][:*NUM*...] (**-groupNames**)

Same as **-g** but print group name for each tag. **-G0** is assumed if *NUM* is not specified. May be combined with a number of other options to add group names to the output. Note that *NUM* may be added wherever **-G** is mentioned in the documentation. See the **-g** option above for details.

-h (**-htmlFormat**)

Use HTML table formatting for output. Implies the **-E** option. The formatting options **-D**, **-H**, **-g**, **-G**, **-I** and **-s** may be used in combination with **-h** to influence the HTML format.

-H (**-hex**)

Show tag ID number in hexadecimal when extracting information.

-htmlDump[*OFFSET*]

Generate a dynamic web page containing a hex dump of the EXIF information. This can be a very powerful tool for low-level analysis of EXIF information. The **-htmlDump** option is also invoked if the **-v** and **-h** options are used together. The verbose level controls the maximum length of the blocks dumped. An *OFFSET* may be given to specify the base for displayed offsets. If not provided, the EXIF/TIFF base offset is used. Use **-htmlDump0** for absolute offsets. Currently only EXIF/TIFF and JPEG information is dumped, but the **-u** option can be used to give a raw hex dump of other file

formats.

-j[+=JSONFILE] (-json)

Use JSON (JavaScript Object Notation) formatting for console output, or import JSON file if *JSONFILE* is specified. This option may be combined with **-g** to organize the output into objects by group, or **-G** to add group names to each tag. List-type tags with multiple items are output as JSON arrays unless **-sep** is used. By default XMP structures are flattened into individual tags in the JSON output, but the original structure may be preserved with the **-struct** option (this also causes all list-type XMP tags to be output as JSON arrays, otherwise single-item lists would be output as simple strings). The **-a** option is implied if the **-g** or **-G** options are used, otherwise it is ignored and tags with identical JSON names are suppressed. (**-g4** may be used to ensure that all tags have unique JSON names.) Adding the **-D** or **-H** option changes tag values to JSON objects with “val” and “id” fields, and adding **-I** adds a “desc” field, and a “num” field if the numerical value is different from the converted “val”. The **-b** option may be added to output binary data, encoded in base64 if necessary (indicated by ASCII “base64:” as the first 7 bytes of the value), and **-t** may be added to include tag table information (see **-t** for details). The JSON output is UTF-8 regardless of any **-L** or **-charset** option setting, but the UTF-8 validation is disabled if a character set other than UTF-8 is specified.

If *JSONFILE* is specified, the file is imported and the tag definitions from the file are used to set tag values on a per-file basis. The special “SourceFile” entry in each JSON object associates the information with a specific target file. An object with a missing SourceFile or a SourceFile of “*” defines default tags for all target files which are combined with any tags specified for the specific SourceFile processed. The imported JSON file must have the same format as the exported JSON files with the exception that the **-g** option is not compatible with the import file format (use **-G** instead). Additionally, tag names in the input JSON file may be suffixed with a # to disable print conversion.

Unlike CSV import, empty values are not ignored, and will cause an empty value to be written if supported by the specific metadata type. Tags are deleted by using the **-f** option and setting the tag value to “-” (or to the MissingTagValue setting if this API option was used). Importing with **-j+=JSONFILE** causes new values to be added to existing lists.

-l (-long)

Use long 2-line Canon-style output format. Adds a description and unconverted value (if it is different from the converted value) to the XML, JSON or PHP output when **-X**, **-j** or **-php** is used. May also be combined with **-listf**, **-listr** or **-listwf** to add descriptions of the file types.

-L (-latin)

Use Windows Latin1 encoding (cp1252) for output tag values instead of the default UTF-8. When writing, **-L** specifies that input text values are Latin1 instead of UTF-8. Equivalent to **-charset latin**.

-lang [LANG]

Set current language for tag descriptions and converted values. *LANG* is *de*, *fr*, *ja*, etc. Use **-lang** with no other arguments to get a list of available languages. The default language is *en* if **-lang** is not specified. Note that tag/group names are always English, independent of the **-lang** setting, and translation of warning/error messages has not yet been implemented. May also be combined with **-listx** to output descriptions in one language only.

By default, ExifTool uses UTF-8 encoding for special characters, but the **-L** or **-charset** option may be used to invoke other encodings. Note that ExifTool uses Unicode::LineBreak if available to help preserve the column alignment of the plain text output for languages with a variable-width character set.

Currently, the language support is not complete, but users are welcome to help improve this by submitting their own translations. To submit a translation, follow these steps (you must have Perl installed for this):

1. Download and unpack the latest Image-ExifTool full distribution.
2. 'cd' into the Image-ExifTool directory.

3. Run this command to make an XML file of the desired tags (eg. EXIF):

```
./exiftool -listx -exif:all > out.xml
```

4. Copy this text into a file called 'import.pl' in the exiftool directory:

```
push @INC, 'lib';
require Image::ExifTool::TagInfoXML;
my $file = shift or die "Expected XML file name\n";
$Image::ExifTool::TagInfoXML::makeMissing = shift;
Image::ExifTool::TagInfoXML::BuildLangModules($file,8);
```

5. Run the 'import.pl' script to Import the XML file, generating the 'MISSING' entries for your language (eg. Russian):

```
perl import.pl out.xml ru
```

6. Edit the generated language module lib/Image/ExifTool/Lang/ru.pm, and search and replace all 'MISSING' strings in the file with your translations.

7. Email the module ('ru.pm' in this example) to philharvey66 at gmail.com

8. Thank you!!

-listItem INDEX

For list-type tags, this causes only the item with the specified index to be extracted. *INDEX* is 0 for the first item in the list. Negative indices may also be used to reference items from the end of the list. Has no effect on single-valued tags. Also applies to tag values when copying from a tag, and in **-if** conditions.

-n (--printConv)

Disable print conversion for all tags. By default, extracted values are converted to a more human-readable format, but the **-n** option disables this conversion, revealing the machine-readable values. For example:

```
> exiftool -Orientation -S a.jpg
Orientation: Rotate 90 CW
> exiftool -Orientation -S -n a.jpg
Orientation: 6
```

The print conversion may also be disabled on a per-tag basis by suffixing the tag name with a # character:

```
> exiftool -Orientation# -Orientation -S a.jpg
Orientation: 6
Orientation: Rotate 90 CW
```

These techniques may also be used to disable the inverse print conversion when writing. For example, the following commands all have the same effect:

```
> exiftool -Orientation='Rotate 90 CW' a.jpg
> exiftool -Orientation=6 -n a.jpg
> exiftool -Orientation#=6 a.jpg
```

-p FMFILE or STR (-printFormat)

Print output in the format specified by the given file or string. The argument is interpreted as a string unless a file of that name exists, in which case the string is loaded from the contents of the file. Tag names in the format file or string begin with a \$ symbol and may contain leading group names and/or a trailing # (to disable print conversion). Case is not significant. Braces { } may be used around the tag name to separate it from subsequent text. Use \$\$ to represent a \$ symbol, and \$/ for a newline.

Multiple **-p** options may be used, each contributing a line (or more) of text to the output. Lines beginning with # [HEAD] and # [TAIL] are output before the first processed file and after the last

processed file respectively. Lines beginning with `#[SECT]` and `#[ENDS]` are output before and after each section of files. A section is defined as a group of consecutive files with the same section header (eg. files are grouped by directory if `#[SECT]` contains `$directory`). Lines beginning with `#[BODY]` and lines not beginning with `#` are output for each processed file. Lines beginning with `#[IF]` are not output, but all BODY lines are skipped if any tag on an IF line doesn't exist. Other lines beginning with `#` are ignored. For example, this format file:

```
# this is a comment line
#[HEAD]-- Generated by ExifTool $exifToolVersion --
File: $FileName - $DateTimeOriginal
(f/$Aperture, ${ShutterSpeed}s, ISO $EXIF:ISO)
#[TAIL]-- end --
```

with this command:

```
exiftool -p test.fmt a.jpg b.jpg
```

produces output like this:

```
-- Generated by ExifTool 12.00 --
File: a.jpg - 2003:10:31 15:44:19
(f/5.6, 1/60s, ISO 100)
File: b.jpg - 2006:05:23 11:57:38
(f/8.0, 1/13s, ISO 100)
-- end --
```

The values of List-type tags with multiple items and Shortcut tags representing multiple tags are joined according to the `-sep` option setting when interpolated in the string.

When `-ee` (`-extractEmbedded`) is combined with `-p`, embedded documents are effectively processed as separate input files.

If a specified tag does not exist, a minor warning is issued and the line with the missing tag is not printed. However, the `-f` option may be used to set the value of missing tags to `'-'` (but this may be configured via the `MissingTagValue` API option), or the `-m` option may be used to ignore minor warnings and leave the missing values empty. Alternatively, `-q -q` may be used to simply suppress the warning messages.

The “Advanced formatting feature” may be used to modify the values of individual tags with the `-p` option.

-php

Format output as a PHP Array. The `-g`, `-G`, `-D`, `-H`, `-I`, `-sep` and `-struct` options combine with `-php`, and duplicate tags are handled in the same way as with the `-json` option. As well, the `-b` option may be added to output binary data, and `-t` may be added to include tag table information (see `-t` for details). Here is a simple example showing how this could be used in a PHP script:

```
<?php
eval('$array=' . `exiftool -php -q image.jpg`);
print_r($array);
?>
```

-s[*NUM*] (-short)

Short output format. Prints tag names instead of descriptions. Add *NUM* or up to 3 `-s` options for even shorter formats:

```
-s1 or -s          - print tag names instead of descriptions
-s2 or -s -s       - no extra spaces to column-align values
-s3 or -s -s -s    - print values only (no tag names)
```

Also effective when combined with `-t`, `-h`, `-X` or `-listx` options.

-S (-veryShort)

Very short format. The same as **-s2** or two **-s** options. Tag names are printed instead of descriptions, and no extra spaces are added to column-align values.

-sep STR (-separator)

Specify separator string for items in list-type tags. When reading, the default is to join list items with “, ”. When writing, this option causes values assigned to list-type tags to be split into individual items at each substring matching *STR* (otherwise they are not split by default). Space characters in *STR* match zero or more whitespace characters in the value.

Note that an empty separator ("") is allowed, and will join items with no separator when reading, or split the value into individual characters when writing.

For pure binary output (**-b** used without **-j**, **-php** or **-X**), the first **-sep** option specifies a list-item separator, and a second **-sep** option specifies a terminator for the end of the list (or after each value if not a list). In these strings, `\n`, `\r` and `\t` may be used to represent a newline, carriage return and tab respectively. By default, binary list items are separated by a newline, and no terminator is added.

-sort, --sort

Sort output by tag description, or by tag name if the **-s** option is used. When sorting by description, the sort order will depend on the **-lang** option setting. Without the **-sort** option, tags appear in the order they were specified on the command line, or if not specified, the order they were extracted from the file. By default, tags are organized by groups when combined with the **-g** or **-G** option, but this grouping may be disabled with **--sort**.

-struct, --struct

Output structured XMP information instead of flattening to individual tags. This option works well when combined with the XML (**-X**) and JSON (**-j**) output formats. For other output formats, XMP structures and lists are serialized into the same format as when writing structured information (see <https://exiftool.org/struct.html> for details). When copying, structured tags are copied by default unless **--struct** is used to disable this feature (although flattened tags may still be copied by specifying them individually unless **-struct** is used). These options have no effect when assigning new values since both flattened and structured tags may always be used when writing.

-t (-tab)

Output a tab-delimited list of description/values (useful for database import). May be combined with **-s** to print tag names instead of descriptions, or **-S** to print tag values only, tab-delimited on a single line. The **-t** option may be combined with **-j**, **-php** or **-X** to add tag table information (`table`, `tag id`, and `index` for cases where multiple conditional tags exist with the same ID).

-T (-table)

Output tag values in table form. Equivalent to **-t -S -q -f**.

-v[*NUM*] (-verbose)

Print verbose messages. *NUM* specifies the level of verbosity in the range 0–5, with higher numbers being more verbose. If *NUM* is not given, then each **-v** option increases the level of verbosity by 1. With any level greater than 0, most other options are ignored and normal console output is suppressed unless specific tags are extracted. Using **-v0** causes the console output buffer to be flushed after each line (which may be useful to avoid delays when piping exiftool output), and prints the name of each processed file when writing. Also see the **-progress** option.

-w[+!]*EXT* or *FMT* (-textOut)

Write console output to files with names ending in *EXT*, one for each source file. The output file name is obtained by replacing the source file extension (including the ‘.’) with the specified extension (and a ‘.’ is added to the start of *EXT* if it doesn’t already contain one). Alternatively, a *FMT* string may be used to give more control over the output file name and directory. In the format string, `%d`, `%f` and `%e` represent the directory, filename and extension of the source file, and `%c` represents a copy number which is automatically incremented if the file already exists. `%d` includes the trailing ‘/’ if necessary, but `%e` does not include the leading ‘.’. For example:

```

-w %d%f.txt      # same effect as "-w txt"
-w dir/%f_%e.out # write files to "dir" as "FILE_EXT.out"
-w dir2/%d%f.txt # write to "dir2", keeping dir structure
-w a%c.txt       # write to "a.txt" or "a1.txt" or "a2.txt"...
```

Existing files will not be changed unless an exclamation point is added to the option name (ie. **-w!** or **-textOut!**) to overwrite the file, or a plus sign (ie. **-w+** or **-textOut+**) to append to the existing file. Both may be used (ie. **-w+!** or **-textOut+!**) to overwrite output files that didn't exist before the command was run, and append the output from multiple source files. For example, to write one output file for all source files in each directory:

```
exiftool -filename -createdate -T -w+! %d/out.txt -r DIR
```

Capitalized format codes %D, %F, %E and %C provide slightly different alternatives to the lower case versions. %D does not include the trailing '/', %F is the full filename including extension, %E includes the leading '.', and %C increments the count for each processed file (see below).

Notes:

1) In a Windows BAT file the % character is represented by %, so an argument like %d%f.txt is written as %%d%%f.txt.

2) If the argument for **-w** does not contain a valid format code (eg. %f), then it is interpreted as a file extension. It is not possible to specify a simple filename as an argument — creating a single output file from multiple source files is typically done by shell redirection, ie)

```
exiftool FILE1 FILE2 ... > out.txt
```

But if necessary, an empty format code may be used to force the argument to be interpreted as a format string, and the same result may be obtained without the use of shell redirection:

```
exiftool -w+! %0fout.txt FILE1 FILE2 ...
```

Advanced features:

A substring of the original file name, directory or extension may be taken by specifying a field width immediately following the '%' character. If the width is negative, the substring is taken from the end. The substring position (characters to ignore at the start or end of the string) may be given by a second optional value after a decimal point. For example:

| Input File Name | Format Specifier | Output File Name |
|------------------|------------------|------------------|
| ----- | ----- | ----- |
| Picture-123.jpg | %7f.txt | Picture.txt |
| Picture-123.jpg | %-.4f.out | Picture.out |
| Picture-123.jpg | %7f.%-3f | Picture.123 |
| Picture-123a.jpg | Meta%-3.1f.txt | Meta123.txt |

(Note that special characters may have a width of greater than one.)

For %d and %D, the field width/position specifiers may be applied to the directory levels instead of substring position by using a colon instead of a decimal point in the format specifier. For example:

| Source Dir | Format | Result | Notes |
|--------------|--------|------------|--------------------------|
| ----- | ----- | ----- | ----- |
| pics/2012/02 | %2:d | pics/2012/ | take top 2 levels |
| pics/2012/02 | %-:1d | pics/2012/ | up one directory level |
| pics/2012/02 | %:1d | 2012/02/ | ignore top level |
| pics/2012/02 | %1:1d | 2012/ | take 1 level after top |
| pics/2012/02 | %-1:D | 02 | bottom level folder name |
| /Users/phil | %:2d | phil/ | ignore top 2 levels |

(Note that the root directory counts as one level when an absolute path is used as in the last example)

above.)

For %c, these modifiers have a different effects. If a field width is given, the copy number is padded with zeros to the specified width. A leading '-' adds a dash before the copy number, and a '+' adds an underline. By default, the copy number is omitted from the first file of a given name, but this can be changed by adding a decimal point to the modifier. For example:

```
-w A%-cZ.txt      # AZ.txt, A-1Z.txt, A-2Z.txt ...
-w B%5c.txt       # B.txt, B00001.txt, B00002.txt ...
-w C%.c.txt       # C0.txt, C1.txt, C2.txt ...
-w D%-c.txt       # D-0.txt, D-1.txt, D-2.txt ...
-w E%-4c.txt      # E-0000.txt, E-0001.txt, E-0002.txt ...
-w F%-4nc.txt     # F-0001.txt, F-0002.txt, F-0003.txt ...
-w G%+c.txt       # G.txt, G_1.txt, G_2.txt ...
-w H%-lc.txt      # H.txt, H-b.txt, H-c.txt ...
-w I.%.3uc.txt    # I.AAA.txt, I.AAB.txt, I.AAC.txt ...
```

A special feature allows the copy number to be incremented for each processed file by using %C (upper case) instead of %c. This allows a sequential number to be added to output file names, even if the names are different. For %C, a copy number of zero is not omitted as it is with %c. A leading '-' causes the number to be reset at the start of each new directory, and '+' has no effect. The number before the decimal place gives the starting index, the number after the decimal place gives the field width. The following examples show the output filenames when used with the command `exiftool`

```
-w %C%f.txt       # 0rose.txt, 1star.txt, 2jet.txt
-w %f-%10C.txt    # rose-10.txt, star-11.txt, jet-12.txt
-w %.3C-%f.txt    # 000-rose.txt, 001-star.txt, 002-jet.txt
-w %57.4C%f.txt   # 0057rose.txt, 0058star.txt, 0059jet.txt
```

All format codes may be modified by 'l' or 'u' to specify lower or upper case respectively (ie. %le for a lower case file extension). When used to modify %c or %C, the numbers are changed to an alphabetical base (see example H above). Also, %c and %C may be modified by 'n' to count using natural numbers starting from 1, instead of 0 (see example F above).

This same *FMT* syntax is used with the `-o` and `-tagsFromFile` options, although %c and %C are only valid for output file names.

-W[+!] *FMT* (-tagOut)

This enhanced version of the `-w` option allows a separate output file to be created for each extracted tag. See the `-w` option documentation above for details of the basic functionality. Listed here are the differences between `-W` and `-w`:

- 1) With `-W`, a new output file is created for each extracted tag.
- 2) `-W` supports three additional format codes: %t, %g and %s represent the tag name, group name, and suggested extension for the output file (based on the format of the data). The %g code may be followed by a single digit to specify the group family number (eg. %g1), otherwise family 0 is assumed. The substring width/position/case specifiers may be used with these format codes in exactly the same way as with %f and %e.
- 3) The argument for `-W` is interpreted as a file name if it contains no format codes. (For `-w`, this would be a file extension.) This change allows a simple file name to be specified, which, when combined with the append feature, provides a method to write metadata from multiple source files to a single output file without the need for shell redirection. For example, the following pairs of commands give the same result:

```
# overwriting existing text file
exiftool test.jpg > out.txt      # shell redirection
exiftool test.jpg -W+! out.txt  # equivalent -W option

# append to existing text file
exiftool test.jpg >> out.txt     # shell redirection
exiftool test.jpg -W+ out.txt    # equivalent -W option
```

4) Adding the **-v** option to **-W** sends a list of the tags and output file names to the console instead of giving a verbose dump of the entire file. (Unless appending all output to one file for each source file by using **-W+** with an output file *FMT* that does not contain %t, %g or %s.)

5) Individual list items are stored in separate files when **-W** is combined with **-b**, but note that for separate files to be created %c or %C must be used in *FMT* to give the files unique names.

-Wext EXT, --Wext EXT (-tagOutExt)

This option is used to specify the type of output file(s) written by the **-W** option. An output file is written only if the suggested extension matches *EXT*. Multiple **-Wext** options may be used to write more than one type of file. Use **--Wext** to write all but the specified type(s).

-X (-xmlFormat)

Use ExifTool-specific RDF/XML formatting for console output. Implies the **-a** option, so duplicate tags are extracted. The formatting options **-b**, **-D**, **-H**, **-I**, **-s**, **-sep**, **-struct** and **-t** may be used in combination with **-X** to affect the output, but note that the tag ID (**-D**, **-H** and **-t**), binary data (**-b**) and structured output (**-struct**) options are not effective for the short output (**-s**). Another restriction of **-s** is that only one tag with a given group and name may appear in the output. Note that the tag ID options (**-D**, **-H** and **-t**) will produce non-standard RDF/XML unless the **-I** option is also used.

By default, **-X** outputs flattened tags, so **-struct** should be added if required to preserve XMP structures. List-type tags with multiple values are formatted as an RDF Bag, but they are combined into a single string when **-s** or **-sep** is used. Using **-L** changes the XML encoding from “UTF-8” to “windows-1252”. Other **-charset** settings change the encoding only if there is a corresponding standard XML character set. The **-b** option causes binary data values to be written, encoded in base64 if necessary. The **-t** option adds tag table information to the output (see **-t** for details).

Note: This output is NOT the same as XMP because it uses dynamically-generated property names corresponding to the ExifTool tag names, and not the standard XMP properties. To write XMP instead, use the **-o** option with an XMP extension for the output file.

Processing control

-a, --a (-duplicates, --duplicates)

Allow (**-a**) or suppress (**--a**) duplicate tag names to be extracted. By default, duplicate tags are suppressed when reading unless the **-ee** or **-X** options are used or the Duplicates option is enabled in the configuration file. This option has an affect when writing only to allow duplicate Warning messages to be shown. Duplicate tags are always extracted when copying.

-e (--composite)

Extract existing tags only — don't generate composite tags.

-ee (-extractEmbedded)

Extract information from embedded documents in EPS files, embedded EPS information and JPEG and Jpeg2000 images in PDF files, embedded MPF images in JPEG and MPO files, streaming metadata in AVCHD videos, and the resource fork of Mac OS files. Implies the **-a** option. Use **-g3** or **-G3** to identify the originating document for extracted information. Embedded documents containing sub-documents are indicated with dashes in the family 3 group name. (eg. Doc2-3 is the 3rd sub-document of the 2nd embedded document.) Note that this option may increase processing time substantially, especially for PDF files with many embedded images or videos with streaming metadata.

When used with **-ee**, the **-p** option is evaluated for each embedded document as if it were a separate

input file. This allows, for example, generation of GPS track logs from timed metadata in videos. See <<https://exiftool.org/geotag.html#Inverse>> for examples.

-ext[+] EXT, --ext EXT (-extension)

Process only files with (**-ext**) or without (**--ext**) a specified extension. There may be multiple **-ext** and **--ext** options. A plus sign may be added (ie. **-ext+**) to add the specified extension to the normally processed files. EXT may begin with a leading '.', which is ignored. Case is not significant. "*" may be used to process files with any extension (or none at all), as in the last three examples:

```
exiftool -ext JPG DIR           # process only JPG files
exiftool --ext cr2 --ext dng DIR # supported files but CR2/DNG
exiftool -ext+ txt DIR          # supported files plus TXT
exiftool -ext "*" DIR           # process all files
exiftool -ext "*" --ext xml DIR  # process all but XML files
exiftool -ext "*" --ext . DIR    # all but those with no ext
```

Using this option has two main advantages over specifying ***.EXT** on the command line: 1) It applies to files in subdirectories when combined with the **-r** option. 2) The **-ext** option is case-insensitive, which is useful when processing files on case-sensitive filesystems.

Note that all files specified on the command line will be processed regardless of extension unless the **-ext** option is used.

-F[OFFSET] (-fixBase)

Fix the base for maker notes offsets. A common problem with some image editors is that offsets in the maker notes are not adjusted properly when the file is modified. This may cause the wrong values to be extracted for some maker note entries when reading the edited file. This option allows an integer *OFFSET* to be specified for adjusting the maker notes base offset. If no *OFFSET* is given, ExifTool takes its best guess at the correct base. Note that exiftool will automatically fix the offsets for images which store original offset information (eg. newer Canon models). Offsets are fixed permanently if **-F** is used when writing EXIF to an image. eg)

```
exiftool -F -exif:resolutionunit=inches image.jpg
```

-fast[*NUM*]

Increase speed of extracting information. With **-fast** (or **-fast1**), ExifTool will not scan to the end of a JPEG image to check for an AFCP or PreviewImage trailer, or past the first comment in GIF images or the audio/video data in WAV/AVI files to search for additional metadata. These speed benefits are small when reading images directly from disk, but can be substantial if piping images through a network connection. For more substantial speed benefits, **-fast2** also causes exiftool to avoid extracting any EXIF MakerNote information. **-fast3** avoids extracting metadata from the file, and returns only pseudo System tags, but still reads the file header to obtain an educated guess at FileType. **-fast4** doesn't even read the file header, and returns only System tags and a FileType based on the file extension. Has no effect when writing.

Note that a separate **-fast** setting may be used for evaluation of a **-if** condition, or when ordering files with the **-fileOrder** option. See the **-if** and **-fileOrder** options for details.

-fileOrder[*NUM*] [-]TAG

Set file processing order according to the sorted value of the specified TAG. For example, to process files in order of date:

```
exiftool -fileOrder DateTimeOriginal DIR
```

Additional **-fileOrder** options may be added for secondary sort keys. Numbers are sorted numerically, and all other values are sorted alphabetically. Files missing the specified tag are sorted last. The sort order may be reversed by prefixing the tag name with a - (eg. **-fileOrder -createdate**). Print conversion of the sorted values is disabled with the **-n** option, or a # appended to the tag name. Other formatting options (eg. **-d**) have no effect on the sorted values. Note that the **-fileOrder** option can have a large performance impact since it involves an additional

processing pass of each file, but this impact may be reduced by specifying a *NUM* for the **-fast** level used during the metadata-extraction phase. For example, **-fileOrder4** may be used if *TAG* is a pseudo System tag. If multiple **-fileOrder** options are used, the extraction is done at the lowest **-fast** level.

-i DIR (-ignore)

Ignore specified directory name. *DIR* may be either an individual folder name, or a full path. If a full path is specified, it must match the Directory tag exactly to be ignored. Use multiple **-i** options to ignore more than one directory name. A special *DIR* value of SYMLINKS (case sensitive) may be specified to ignore symbolic links when the **-r** option is used.

-if[*NUM*] *EXPR*

Specify a condition to be evaluated before processing each *FILE*. *EXPR* is a Perl-like logic expression containing tag names prefixed by \$ symbols. It is evaluated with the tags from each *FILE* in turn, and the file is processed only if the expression returns true. Unlike Perl variable names, tag names are not case sensitive and may contain a hyphen. As well, tag names may have a leading group names separated by colons, and/or a trailing # character to disable print conversion. The expression \$GROUP:all evaluates to 1 if any tag exists in the specified GROUP, or 0 otherwise (see note 2 below). When multiple **-if** options are used, all conditions must be satisfied to process the file. Returns an exit status of 2 if all files fail the condition. Below are a few examples:

```
# extract shutterspeed from all Canon images in a directory
exiftool -shutterspeed -if '$make eq "Canon"' dir

# add one hour to all images created on or after Apr. 2, 2006
exiftool -alldates+=1 -if '$CreateDate ge "2006:04:02"' dir

# set EXIF ISO value if possible, unless it is set already
exiftool '-exif:iso<iso' -if 'not $exif:iso' dir

# find images containing a specific keyword (case insensitive)
exiftool -if '$keywords =~ /harvey/i' -filename dir
```

Adding *NUM* to the **-if** option causes a separate processing pass to be executed for evaluating *EXPR* at a **-fast** level given by *NUM* (see the **-fast** option documentation for details). Without *NUM*, only one processing pass is done at the level specified by the **-fast** option. For example, using **-if4** is possible if *EXPR* uses only pseudo System tags, and may significantly speed processing if enough files fail the condition.

The expression has access to the current ExifTool object through \$self, and the following special functions are available to allow short-circuiting of the file processing. Both functions have a return value of 1. Case is significant for function names.

```
End()      - end processing after this file
EndDir()   - end processing of files in this directory
```

Notes:

- 1) The **-n** and **-b** options also apply to tags used in *EXPR*.
- 2) Some binary data blocks are not extracted unless specified explicitly. These tags are not available for use in the **-if** condition unless they are also specified on the command line. The alternative is to use the \$GROUP:all syntax. (eg. Use \$exif:all instead of \$exif in *EXPR* to test for the existence of EXIF tags.)
- 3) Tags in the string are interpolated the same way as with **-p** before the expression is evaluated. In this interpolation, \$/ is converted to a newline and \$\$ represents a single \$ symbol (so Perl variables, if used, require a double \$).
- 4) The condition may only test tags from the file being processed. To process one file based on tags

from another, two steps are required. For example, to process XMP sidecar files in directory *DIR* based on tags from the associated NEF:

```
exiftool -if EXPR -p '$directory/$filename' -ext nef DIR > nef.txt
exiftool -@ nef.txt -srcfile %d%f.xmp ...
```

5) The **-a** option has no effect on the evaluation of the expression, and the values of duplicate tags are accessible only by specifying a group name (such as a family 4 instance number, eg. *\$Copy1:TAG*, *\$Copy2:TAG*, etc).

6) A special “OK” UserParam is available to test the success of the previous command when **-execute** was used, and may be used like any other tag in the condition (ie. “\$OK”).

-m (-ignoreMinorErrors)

Ignore minor errors and warnings. This enables writing to files with minor errors and disables some validation checks which could result in minor warnings. Generally, minor errors/warnings indicate a problem which usually won’t result in loss of metadata if ignored. However, there are exceptions, so ExifTool leaves it up to you to make the final decision. Minor errors and warnings are indicated by “[minor]” at the start of the message. Warnings which affect processing when ignored are indicated by “[Minor]” (with a capital “M”). Note that this causes missing values in **-tagsFromFile**, **-p** and **-if** strings to be set to an empty string rather than an undefined value.

-o OUTFILE or FMT (-out)

Set the output file or directory name when writing information. Without this option, when any “real” tags are written the original file is renamed to *FILE_original* and output is written to *FILE*. When writing only FileName and/or Directory “pseudo” tags, **-o** causes the file to be copied instead of moved, but directories specified for either of these tags take precedence over that specified by the **-o** option.

OUTFILE may be **-** to write to stdout. The output file name may also be specified using a *FMT* string in which *%d*, *%f* and *%e* represent the directory, file name and extension of *FILE*. Also, *%c* may be used to add a copy number. See the **-w** option for *FMT* string examples.

The output file is taken to be a directory name if it already exists as a directory or if the name ends with *’/’*. Output directories are created if necessary. Existing files will not be overwritten. Combining the **-overwrite_original** option with **-o** causes the original source file to be erased after the output file is successfully written.

A special feature of this option allows the creation of certain types of files from scratch, or with the metadata from another type of file. The following file types may be created using this technique:

```
XMP, EXIF, EXV, MIE, ICC/ICM, VRD, DR4
```

The output file type is determined by the extension of *OUTFILE* (specified as **- .EXT** when writing to stdout). The output file is then created from a combination of information in *FILE* (as if the **-tagsFromFile** option was used), and tag values assigned on the command line. If no *FILE* is specified, the output file may be created from scratch using only tags assigned on the command line.

-overwrite_original

Overwrite the original *FILE* (instead of preserving it by adding *_original* to the file name) when writing information to an image. Caution: This option should only be used if you already have separate backup copies of your image files. The overwrite is implemented by renaming a temporary file to replace the original. This deletes the original file and replaces it with the edited version in a single operation. When combined with **-o**, this option causes the original file to be deleted if the output file was successfully written (ie. the file is moved instead of copied).

-overwrite_original_in_place

Similar to **-overwrite_original** except that an extra step is added to allow the original file attributes to be preserved. For example, on a Mac this causes the original file creation date, type, creator, label color, icon, Finder tags, other extended attributes and hard links to the file to be preserved (but note that the Mac OS resource fork is always preserved unless specifically deleted with **-rsrc:all=**).

This is implemented by opening the original file in update mode and replacing its data with a copy of a temporary file before deleting the temporary. The extra step results in slower performance, so the **-overwrite_original** option should be used instead unless necessary.

Note that this option reverts to the behaviour of the **-overwrite_original** option when also writing the FileName and/or Directory tags.

-P (-preserve)

Preserve the filesystem modification date/time (FileModifyDate) of the original file when writing. Note that some filesystems store a creation date (ie. FileCreateDate on Windows and Mac systems) which is not affected by this option. This creation date is preserved on Windows systems where Win32API::File and Win32::API are available regardless of this setting. For other systems, the **-overwrite_original_in_place** option may be used if necessary to preserve the creation date. The **-P** option is superseded by any value written to the FileModifyDate tag.

-password *PASSWD*

Specify password to allow processing of password-protected PDF documents. If a password is required but not given, a warning is issued and the document is not processed. This option is ignored if a password is not required.

-progress[:*[TITLE]*]

Show the progress when processing files. Without a colon, the **-progress** option adds a progress count in brackets after the name of each processed file, giving the current file number and the total number of files to be processed. Implies the **-v0** option, causing the names of processed files to also be printed when writing. When combined with the **-if** option, the total count includes all files before the condition is applied, but files that fail the condition will not have their names printed.

If followed by a colon (ie. **-progress:**), the console window title is set according to the specified *TITLE* string. If no *TITLE* is given, a default *TITLE* string of "ExifTool %p%%" is assumed. In the string, %f represents the file name, %p is the progress as a percent, %r is the progress as a ratio, %##b is a progress bar of width "##" (20 characters if "##" is omitted), and %% is a % character. May be combined with the normal **-progress** option to also show the progress count in console messages. (Note: For this feature to function correctly on Mac/Linux, stderr must go to the console.)

-q (-quiet)

Quiet processing. One **-q** suppresses normal informational messages, and a second **-q** suppresses warnings as well. Error messages can not be suppressed, although minor errors may be downgraded to warnings with the **-m** option, which may then be suppressed with **-q -q**.

-r[.] (-recurse)

Recursively process files in subdirectories. Only meaningful if *FILE* is a directory name. Subdirectories with names beginning with "." are not processed unless "." is added to the option name (ie. **-r.** or **-recurse.**). By default, exiftool will also follow symbolic links to directories if supported by the system, but this may be disabled with **-i SYMLINKS** (see the **-i** option for details). Combine this with **-ext** options to control the types of files processed.

-scanForXMP

Scan all files (even unsupported formats) for XMP information unless found already. When combined with the **-fast** option, only unsupported file types are scanned. Warning: It can be time consuming to scan large files.

-u (-unknown)

Extract values of unknown tags. Add another **-u** to also extract unknown information from binary data blocks. This option applies to tags with numerical tag ID's, and causes tag names like "Exif_0xc5d9" to be generated for unknown information. It has no effect on information types which have human-readable tag ID's (such as XMP), since unknown tags are extracted automatically from these formats.

-U (-unknown2)

Extract values of unknown tags as well as unknown information from some binary data blocks. This is the same as two **-u** options.

-wm *MODE* (-writeMode)

Set mode for writing/creating tags. *MODE* is a string of one or more characters from the list below. The default write mode is *wcg*.

```
w - Write existing tags
c - Create new tags
g - create new Groups as necessary
```

For example, use **-wm cg** to only create new tags (and avoid editing existing ones).

The level of the group is the SubDirectory level in the metadata structure. For XMP or IPTC this is the full XMP/IPTC block (the family 0 group), but for EXIF this is the individual IFD (the family 1 group).

-z (-zip)

When reading, causes information to be extracted from .gz and .bz2 compressed images (only one image per archive; requires gzip and bzip2 to be available). When writing, causes compressed information to be written if supported by the metadata format (eg. compressed textual metadata in PNG), disables the recommended padding in embedded XMP (saving 2424 bytes when writing XMP in a file), and writes XMP in shorthand format — the equivalent of setting the API Compress=1 and Compact="NoPadding,Shorthand".

*Other options***-@ *ARGFILE***

Read command-line arguments from the specified file. The file contains one argument per line (NOT one option per line — some options require additional arguments, and all arguments must be placed on separate lines). Blank lines and lines beginning with # are ignored (unless they start with # [CSTR], in which case the rest of the line is treated as a C string, allowing standard C escape sequences such as "\n" for a newline). White space at the start of a line is removed. Normal shell processing of arguments is not performed, which among other things means that arguments should not be quoted and spaces are treated as any other character. *ARGFILE* may exist relative to either the current directory or the exiftool directory unless an absolute pathname is given.

For example, the following *ARGFILE* will set the value of Copyright to "Copyright YYYY, Phil Harvey", where "YYYY" is the year of CreateDate:

```
-d
%Y
-copyright<Copyright $createdate, Phil Harvey
```

Arguments in *ARGFILE* behave exactly the same as if they were entered at the location of the **-@** option on the command line, with the exception that the **-config** and **-common_args** options may not be used in an *ARGFILE*.

-k (-pause)

Pause with the message `-- press any key --` or `-- press RETURN --` (depending on your system) before terminating. This option is used to prevent the command window from closing when run as a Windows drag and drop application.

-list, -listw, -listf, -listr, -listwf, -listg[*NUM*], -listd, -listx

Print a list of all valid tag names (**-list**), all writable tag names (**-listw**), all supported file extensions (**-listf**), all recognized file extensions (**-listr**), all writable file extensions (**-listwf**), all tag groups [in a specified family] (**-listg[*NUM*]**), all deletable tag groups (**-listd**), or an XML database of tag details including language translations (**-listx**). The **-list**, **-listw** and **-listx** options may be followed by an additional argument of the form **-GROUP:All** to list only tags in a specific group, where *GROUP* is one or more family 0–2 group names (excepting EXIF IFD groups) separated by colons. With **-listg**,

NUM may be given to specify the group family, otherwise family 0 is assumed. The **-l** option may be combined with **-listf**, **-listr** or **-listwf** to add file descriptions to the list. The **-lang** option may be combined with **-listx** to output descriptions in a single language. Here are some examples:

```
-list          # list all tag names
-list -EXIF:All # list all EXIF tags
-list -xmp:time:all # list all XMP tags relating to time
-listw -XMP-dc:All # list all writable XMP-dc tags
-listf        # list all supported file extensions
-listr        # list all recognized file extensions
-listwf       # list all writable file extensions
-listg1       # list all groups in family 1
-listd        # list all deletable groups
-listx -EXIF:All # list database of EXIF tags in XML format
-listx -XMP:All -s # list short XML database of XMP tags
```

When combined with **-listx**, the **-s** option shortens the output by omitting the descriptions and values (as in the last example above), and **-f** adds a 'flags' attribute if applicable. The flags are formatted as a comma-separated list of the following possible values: Avoid, Binary, List, Mandatory, Permanent, Protected, Unknown and Unsafe (see the Tag Name documentation). For XMP List tags, the list type (Alt, Bag or Seq) is added to the flags, and flattened structure tags are indicated by a Flattened flag.

Note that none of the **-list** options require an input *FILE*.

- ver** Print exiftool version number. The **-v** option may be added to print addition system information (see the README file of the full distribution for more details about optional libraries), or **-v2** to also list the Perl include directories.
- Indicates the end of options. Any remaining arguments are treated as file names, even if they begin with a dash (-).

Special features

-geotag *TRKFILE*

Geotag images from the specified GPS track log file. Using the **-geotag** option is equivalent to writing a value to the Geotag tag. The GPS position is interpolated from the track at a time specified by the value written to the Geotime tag. If Geotime is not specified, the value is copied from DateTimeOriginal# (the # is added to copy the unformatted value, avoiding potential conflicts with the **-d** option). For example, the following two commands are equivalent:

```
exiftool -geotag trk.log image.jpg
exiftool -geotag trk.log "-Geotime<DateTimeOriginal#" image.jpg
```

When the `Geotime` value is converted to UTC, the local system timezone is assumed unless the date/time value contains a timezone. Writing `Geotime` causes the following tags to be written (provided they can be calculated from the track log, and they are supported by the destination metadata format): `GPSPLatitude`, `GPSPLatitudeRef`, `GPSPLongitude`, `GPSPLongitudeRef`, `GPSPAltitude`, `GPSPAltitudeRef`, `GPSPDateStamp`, `GPSPTimeStamp`, `GPSPDateTime`, `GPSPTrack`, `GPSPTrackRef`, `GPSSpeed`, `GPSSpeedRef`, `GPSPImgDirection`, `GPSPImgDirectionRef`, `GPSPitch`, `GPSPRoll`, `AmbientTemperature` and `CameraElevationAngle`. By default, tags are created in EXIF, and updated in XMP only if they already exist. However, `EXIF:Geotime` or `XMP:Geotime` may be specified to write only EXIF or XMP tags respectively. Note that `GPSPitch` and `GPSPRoll` are non-standard, and require user-defined tags in order to be written.

The `Geosync` tag may be used to specify a time correction which is applied to each `Geotime` value for synchronization with GPS time. For example, the following command compensates for image times which are 1 minute and 20 seconds behind GPS:

```
exiftool -geosync=+1:20 -geotag a.log DIR
```

Advanced Geosync features allow a linear time drift correction and synchronization from previously geotagged images. See “geotag.html” in the full ExifTool distribution for more information.

Multiple **-geotag** options may be used to concatenate GPS track log data. Also, a single **-geotag** option may be used to load multiple track log files by using wildcards in the *TRKFILE* name, but note that in this case *TRKFILE* must be quoted on most systems (with the notable exception of Windows) to prevent filename expansion. For example:

```
exiftool -geotag "TRACKDIR/*.log" IMAGEDIR
```

Currently supported track file formats are GPX, NMEA RMC/GGA/GLL, KML, IGC, Garmin XML and TCX, Magellan PMGNTRK, Honeywell PTNTHPR, Bramor gEO, Winplus Beacon TXT, and GPS/IMU CSV files. See “GEOTAGGING EXAMPLES” for examples. Also see “geotag.html” in the full ExifTool distribution and the Image::ExifTool Options for more details and for information about geotag configuration options.

-globalTimeShift *SHIFT*

Shift all formatted date/time values by the specified amount when reading. Does not apply to unformatted (**-n**) output. *SHIFT* takes the same form as the date/time shift when writing (see Image::ExifTool::Shift.pl for details), with a negative shift being indicated with a minus sign (-) at the start of the *SHIFT* string. For example:

```
# return all date/times, shifted back by 1 hour
exiftool -globalTimeShift -1 -time:all a.jpg

# set the file name from the shifted CreateDate (-1 day) for
# all images in a directory
exiftool "-filename<createdate" -globaltimeshift "-0:0:1 0:0:0" \
-d %Y%m%d-%H%M%S.%e dir
```

-use *MODULE*

Add features from specified plug-in *MODULE*. Currently, the MWG module is the only plug-in module distributed with exiftool. This module adds read/write support for tags as recommended by the Metadata Working Group. As a convenience, **-use MWG** is assumed if the MWG group is specified for any tag on the command line. See the MWG Tags documentation for more details. Note that this option is not reversible, and remains in effect until the application terminates, even across the **-execute** option.

Utilities

-restore_original

-delete_original [**!**]

These utility options automate the maintenance of the *_original* files created by exiftool. They have no effect on files without an *_original* copy. The **-restore_original** option restores the specified files from their original copies by renaming the *_original* files to replace the edited versions. For example, the following command restores the originals of all JPG images in directory DIR:

```
exiftool -restore_original -ext jpg DIR
```

The **-delete_original** option deletes the *_original* copies of all files specified on the command line. Without a trailing **!** this option prompts for confirmation before continuing. For example, the following command deletes *a.jpg_original* if it exists, after asking “Are you sure?”:

```
exiftool -delete_original a.jpg
```

These options may not be used with other options to read or write tag values in the same command, but may be combined with options such **-ext**, **-if**, **-r**, **-q** and **-v**.

Advanced options

Among other things, the advanced options allow complex processing to be performed from a single command without the need for additional scripting. This may be particularly useful for implementations such as Windows drag-and-drop applications. These options may also be used to improve performance in multi-pass processing by reducing the overhead required to load exiftool for each invocation.

-api *OPT*[*^[^]=[VAL]*]

Set ExifTool API option. *OPT* is an API option name. The option value is set to 1 if *VAL* is omitted. If *VAL* is omitted, the option value is set to undef if = is used, or an empty string with ^=. See Image::ExifTool Options for a list of available API options. This overrides API options set via the config file.

-common_args

Specifies that all arguments following this option are common to all executed commands when **-execute** is used. This and the **-config** option are the only options that may not be used inside a **-@ARGFILE**. Note that by definition this option and its arguments **MUST** come after all other options on the command line.

-config *CFGFILE*

Load specified configuration file instead of the default ".ExifTool_config". If used, this option must come before all other arguments on the command line and applies to all **-execute**'d commands. The *CFGFILE* must exist relative to the current working directory or the exiftool application directory unless an absolute path is specified. Loading of the default config file may be disabled by setting *CFGFILE* to an empty string (ie. ""). See <<https://exiftool.org/config.html>> and config_files/example.config in the full ExifTool distribution for details about the configuration file syntax.

-echo[*NUM*] *TEXT*

Echo *TEXT* to stdout (**-echo** or **-echo1**) or stderr (**-echo2**). Text is output as the command line is parsed, before the processing of any input files. *NUM* may also be 3 or 4 to output text (to stdout or stderr respectively) after processing is complete.

-efile[*NUM*][!] *ERRFILE*

Save the names of files giving errors (*NUM* missing or 1), files that were unchanged (*NUM* is 2), files that fail the **-if** condition (*NUM* is 4), or any combination thereof (by summing *NUM*, eg. **-efile3** is the same as having both **-efile** and **-efile2** options with the same *ERRFILE*). By default, file names are appended to any existing *ERRFILE*, but *ERRFILE* is overwritten if an exclamation point is added to the option (eg. **-efile!**). Saves the name of the file specified by the **-srcfile** option if applicable.

-execute[*NUM*]

Execute command for all arguments up to this point on the command line (plus any arguments specified by **-common_args**). The result is as if the commands were executed as separate command lines (with the exception of the **-config** and **-use** options which remain in effect for subsequent commands). Allows multiple commands to be executed from a single command line. *NUM* is an optional number that is echoed in the "{ready}" message when using the **-stay_open** feature.

-srcfile *FMT*

Specify a different source file to be processed based on the name of the original *FILE*. This may be useful in some special situations for processing related preview images or sidecar files. See the **-w** option for a description of the *FMT* syntax. Note that file name *FMT* strings for all options are based on the original *FILE* specified from the command line, not the name of the source file specified by **-srcfile**.

For example, to copy metadata from NEF files to the corresponding JPG previews in a directory where other JPG images may exist:

```
exiftool -ext nef -tagsfromfile @ -srcfile %d%f.jpg dir
```

If more than one **-srcfile** option is specified, the files are tested in order and the first existing source file is processed. If none of the source files already exist, then exiftool uses the first **-srcfile**

specified.

A *FMT* of @ may be used to represent the original *FILE*, which may be useful when specifying multiple **-srcfile** options (eg. to fall back to processing the original *FILE* if no sidecar exists).

When this option is used, two special UserParam tags (OriginalFileName and OriginalDirectory) are generated to allow access to the original *FILE* name and directory.

-stay_open *FLAG*

If *FLAG* is 1 or True, causes exiftool keep reading from the **-@** *ARGFILE* even after reaching the end of file. This feature allows calling applications to pre-load exiftool, thus avoiding the overhead of loading exiftool for each command. The procedure is as follows:

- 1) Execute `exiftool -stay_open True -@ ARGFILE`, where *ARGFILE* is the name of an existing (possibly empty) argument file or - to pipe arguments from the standard input.
- 2) Write exiftool command-line arguments to *ARGFILE*, one argument per line (see the **-@** option for details).
- 3) Write `-execute\n` to *ARGFILE*, where \n represents a newline sequence. (Note: You may need to flush your write buffers here if using buffered output.) Exiftool will then execute the command with the arguments received up to this point, send a "{ready}" message to stdout when done (unless the **-q** or **-T** option is used), and continue trying to read arguments for the next command from *ARGFILE*. To aid in command/response synchronization, any number appended to the `-execute` option is echoed in the "{ready}" message. For example, `-execute613` results in "{ready613}".
- 4) Repeat steps 2 and 3 for each command.
- 5) Write `-stay_open\nFalse\n` to *ARGFILE* when done. This will cause exiftool to process any remaining command-line arguments then exit normally.

The input *ARGFILE* may be changed at any time before step 5 above by writing the following lines to the currently open *ARGFILE*:

```
-stay_open
True
-@
NEWARGFILE
```

This causes *ARGFILE* to be closed, and *NEWARGFILE* to be kept open. (Without the **-stay_open** here, exiftool would have returned to reading arguments from *ARGFILE* after reaching the end of *NEWARGFILE*.)

Note: When writing arguments to a disk file there is a delay of up to 0.01 seconds after writing `-execute\n` before exiftool starts processing the command. This delay may be avoided by sending a CONT signal to the exiftool process immediately after writing `-execute\n`. (There is no associated delay when writing arguments via a pipe with **-@** -, so the signal is not necessary when using this technique.)

-userParam *PARAM*[[^]=[*VAL*]]

Set user parameter. *PARAM* is an arbitrary user parameter name. This is an interface to the API UserParam option (see the Image::ExifTool Options documentation), and provides a method to access user-defined parameters in arguments to the **-if** and **-p** options as if they were any other tag. Appending a hash tag (#) to *PARAM* also causes the parameter to be extracted as a normal tag (in the UserParam group). Similar to the **-api** option, the parameter value is set to 1 if *=VAL* is omitted, undef if just *VAL* is omitted with =, or an empty string if *VAL* is omitted with ^=.

```
exiftool -p '$test from $filename' -userparam test=Hello FILE
```

Advanced formatting feature

An advanced formatting feature allows modification of the value of any tag interpolated within a **-if** or **-p** option argument, or a **-tagsFromFile** redirection string. Tag names within these strings are prefixed by a \$

symbol, and an arbitrary Perl expression may be applied to the tag value by placing braces around the tag name and inserting the expression after the name, separated by a semicolon (ie. `${TAG;EXPR}`). The expression acts on the value of the tag through the default input variable (`$_`), and has access to the full ExifTool API through the current ExifTool object (`$self`) and the tag key (`$tag`). It may contain any valid Perl code, including translation (`tr///`) and substitution (`s///`) operations, but note that braces within the expression must be balanced. The example below prints the camera Make with spaces translated to underlines, and multiple consecutive underlines replaced by a single underline:

```
exiftool -p '${make;tr/ /_/;s/___+/_/g}' image.jpg
```

An `@` may be added after the tag name to make the expression act on individual list items for list-type tags, simplifying list processing. Set `$_` to `undef` to remove an item from the list. As an example, the following command returns all subjects not containing the string "xxx":

```
exiftool -p '${subject@;$_=undef if /xxx/}' image.jpg
```

A default expression of `tr(/\\?*:|"<>\0)()d` is assumed if the expression is empty (ie. `${TAG;}`). This removes the characters `/\?*:|<>` and null from the printed value. (These characters are illegal in Windows file names, so this feature is useful if tag values are used in file names.)

Helper functions

DateFmt

Simplifies reformatting of individual date/time values. This function acts on a standard EXIF-formatted date/time value in `$_` and formats it according to the specified format string (see the `-d` option). To avoid trying to reformat an already-formatted date/time value, a `#` must be added to the tag name (as in the example below) if the `-d` option is also used. For example:

```
exiftool -p '${createdate#;DateFmt("%Y-%m-%d_%H%M%S")}' a.jpg
```

ShiftTime

Shifts EXIF-formatted date/time string by a specified amount. Start with a leading minus sign to shift backwards in time. See `Image::ExifTool::Shift.pl` for details about shift syntax. For example, to shift a date/time value back by one year:

```
exiftool -p '${createdate;ShiftTime("-1:0:0 0")}' a.jpg
```

NoDups

Removes duplicate items from a list with a separator specified by the `-sep` option. This function is most useful when copying list-type tags. For example, the following command may be used to remove duplicate Keywords:

```
exiftool -sep '##' '-keywords<${keywords;NoDups}' a.jpg
```

The `-sep` option is necessary to split the string back into individual list items when writing to a list-type tag.

An optional flag argument may be set to 1 to cause `NoDups` to return `undef` if no duplicates existed, thus preventing the file from being rewritten unnecessarily:

```
exiftool -sep '##' '-keywords<${keywords;NoDups(1)}' a.jpg
```

Note that function names are case sensitive.

WINDOWS UNICODE FILE NAMES

In Windows, command-line arguments are specified using the current code page and are recoded automatically to the system code page. This recoding is not done for arguments in ExifTool arg files, so by default filenames in arg files use the system code page. Unfortunately, these code pages are not complete character sets, so not all file names may be represented.

ExifTool 9.79 and later allow the file name encoding to be specified with `-charset filename=CHARSET`, where `CHARSET` is the name of a valid ExifTool character set, preferably UTF8 (see the `-charset` option for a complete list). Setting this triggers the use of Windows wide-character i/o

routines, thus providing support for most Unicode file names (see note 4). But note that it is not trivial to pass properly encoded file names on the Windows command line (see <<https://exiftool.org/faq.html#Q18>> for details), so placing them in a UTF-8 encoded `-@ argfile` and using `-charset filename=utf8` is recommended if possible.

A warning is issued if a specified filename contains special characters and the filename character set was not provided. However, the warning may be disabled by setting `-charset filename=""`, and ExifTool may still function correctly if the system code page matches the character set used for the file names.

When a directory name is provided, the file name encoding need not be specified (unless the directory name contains special characters), and ExifTool will automatically use wide-character routines to scan the directory.

The filename character set applies to the *FILE* arguments as well as filename arguments of `-@`, `-geotag`, `-o`, `-p`, `-srcfile`, `-tagsFromFile`, `-csv=`, `-j=` and `-TAG<=`. However, it does not apply to the `-config filename`, which always uses the system character set. The `-charset filename=` option must come before the `-@` option to be effective, but the order doesn't matter with respect to other options.

Notes:

- 1) FileName and Directory tag values still use the same encoding as other tag values, and are converted to/from the filename character set when writing/reading if specified.
- 2) Unicode support is not yet implemented for other Windows-based systems like Cygwin.
- 3) See "WRITING READ-ONLY FILES" below for a note about editing read-only files with Unicode names.
- 4) Unicode file names with surrogate pairs (code points over U+FFFF) still cause problems.

WRITING READ-ONLY FILES

In general, ExifTool may be used to write metadata to read-only files provided that the user has write permission in the directory. However, there are three cases where file write permission is also required:

- 1) When using the `-overwrite_original_in_place` option.
- 2) When writing only pseudo System tags (eg. FileModifyDate).
- 3) On Windows if the file has Unicode characters in its name, and a) the `-overwrite_original` option is used, or b) the `_original` backup already exists.

Hidden files in Windows behave as read-only files when attempting to write any real tags to the file — an error is generated when using the `-overwrite_original_in_place`, otherwise writing should be successful and the hidden attribute will be removed. But the `-if` option may be used to avoid processing hidden files (provided Win32API::File is available):

```
exiftool -if "$fileattributes !~ /Hidden/" ...
```

READING EXAMPLES

Note: Beware when cutting and pasting these examples into your terminal! Some characters such as single and double quotes and hyphens may have been changed into similar-looking yet functionally-different characters by the text formatter used to display this documentation. Also note that Windows users must use double quotes instead of single quotes as below around arguments containing special characters.

```
exiftool -a -u -g1 a.jpg
```

Print all meta information in an image, including duplicate and unknown tags, sorted by group (for family 1). For performance reasons, this command may not extract all available metadata. (Metadata in embedded documents, metadata extracted by external utilities, and metadata requiring excessive processing time may not be extracted). Add `-ee` and `-api RequestAll=3` to the command to extract absolutely everything available.

```
exiftool -common dir
```

Print common meta information for all images in `dir`. `-common` is a shortcut tag representing common EXIF meta information.

```

exiftool -T -createdate -aperture -shutterspeed -iso dir > out.txt
    List specified meta information in tab-delimited column form for all images in dir to an output text
    file named "out.txt".

exiftool -s -ImageSize -ExposureTime b.jpg
    Print ImageSize and ExposureTime tag names and values.

exiftool -l -canon c.jpg d.jpg
    Print standard Canon information from two image files.

exiftool -r -w .txt -common pictures
    Recursively extract common meta information from files in pictures directory, writing text output
    to .txt files with the same names.

exiftool -b -ThumbnailImage image.jpg > thumbnail.jpg
    Save thumbnail image from image.jpg to a file called thumbnail.jpg.

exiftool -b -JpgFromRaw -w _JFR.JPG -ext NEF -r .
    Recursively extract JPG image from all Nikon NEF files in the current directory, adding _JFR.JPG
    for the name of the output JPG files.

exiftool -a -b -W %d%f_%t%-c.%s -preview:all dir
    Extract all types of preview images (ThumbnailImage, PreviewImage, JpgFromRaw, etc.) from files
    in directory "dir", adding the tag name to the output preview image file names.

exiftool -d '%r %a, %B %e, %Y' -DateTimeOriginal -S -s -ext jpg .
    Print formatted date/time for all JPG files in the current directory.

exiftool -IFD1:XResolution -IFD1:YResolution image.jpg
    Extract image resolution from EXIF IFD1 information (thumbnail image IFD).

exiftool '*resolution*' image.jpg
    Extract all tags with names containing the word "Resolution" from an image.

exiftool -xmp:author:all -a image.jpg
    Extract all author-related XMP information from an image.

exiftool -xmp -b a.jpg > out.xmp
    Extract complete XMP data record intact from a.jpg and write it to out.xmp using the special
    XMP tag (see the Extra tags in Image::ExifTool::TagNames).

exiftool -p '$filename has date $dateTimeOriginal' -q -f dir
    Print one line of output containing the file name and DateTimeOriginal for each image in directory
    dir.

exiftool -ee -p '$gpslatitude, $gpslongitude, $gpstimestamp' a.m2ts
    Extract all GPS positions from an AVCHD video.

exiftool -icc_profile -b -w icc image.jpg
    Save complete ICC_Profile from an image to an output file with the same name and an extension of
    .icc.

exiftool -htmldump -w tmp/%f_%e.html t/images
    Generate HTML pages from a hex dump of EXIF information in all images from the t/images
    directory. The output HTML files are written to the tmp directory (which is created if it didn't exist),
    with names of the form 'FILENAME_EXT.html'.

exiftool -a -b -ee -embeddedimage -W Image_%.3g3.%s file.pdf
    Extract embedded JPG and JP2 images from a PDF file. The output images will have file names like
    "Image_#.jpg" or "Image_#.jp2", where "#" is the ExifTool family 3 embedded document number
    for the image.

```

WRITING EXAMPLES

Note that quotes are necessary around arguments which contain certain special characters such as `>`, `<` or any white space. These quoting techniques are shell dependent, but the examples below will work for most

Unix shells. With the Windows cmd shell however, double quotes should be used (eg. `-Comment="This is a new comment"`).

`exiftool -Comment='This is a new comment' dst.jpg`

Write new comment to a JPG image (replaces any existing comment).

`exiftool -comment= -o newdir -ext jpg .`

Remove comment from all JPG images in the current directory, writing the modified images to a new directory.

`exiftool -keywords=EXIF -keywords=editor dst.jpg`

Replace existing keyword list with two new keywords (EXIF and editor).

`exiftool -Keywords+=word -o newfile.jpg src.jpg`

Copy a source image to a new file, and add a keyword (word) to the current list of keywords.

`exiftool -exposurecompensation+=-0.5 a.jpg`

Decrement the value of ExposureCompensation by 0.5 EV. Note that += with a negative value is used for decrementing because the -= operator is used for conditional deletion (see next example).

`exiftool -credit==xxx dir`

Delete Credit information from all files in a directory where the Credit value was xxx.

`exiftool -xmp:description-de='kühl' -E dst.jpg`

Write alternate language for XMP:Description, using HTML character escaping to input special characters.

`exiftool -all= dst.jpg`

Delete all meta information from an image. Note: You should NOT do this to RAW images (except DNG) since proprietary RAW image formats often contain information in the makernotes that is necessary for converting the image.

`exiftool -all= -comment='lonely' dst.jpg`

Delete all meta information from an image and add a comment back in. (Note that the order is important: `-comment='lonely' -all=` would also delete the new comment.)

`exiftool -all= --jif:all dst.jpg`

Delete all meta information except JFIF group from an image.

`exiftool -Photoshop:All= dst.jpg`

Delete Photoshop meta information from an image (note that the Photoshop information also includes IPTC).

`exiftool -r -XMP-crss:all= DIR`

Recursively delete all XMP-crss information from images in a directory.

`exiftool '-ThumbnailImage<=thumb.jpg' dst.jpg`

Set the thumbnail image from specified file (Note: The quotes are necessary to prevent shell redirection).

`exiftool '-JpgFromRaw<=%d%f_JFR.JPG' -ext NEF -r .`

Recursively write JPEG images with filenames ending in `_JFR.JPG` to the JpgFromRaw tag of like-named files with extension `.NEF` in the current directory. (This is the inverse of the `-JpgFromRaw` command of the "READING EXAMPLES" section above.)

`exiftool -DateTimeOriginal==0:0:0 1:30:0' dir`

Adjust original date/time of all images in directory `dir` by subtracting one hour and 30 minutes. (This is equivalent to `-DateTimeOriginal=-1.5`. See `Image::ExifTool::Shift.pl` for details.)

`exiftool -createdate+=3 -modifydate+=3 a.jpg b.jpg`

Add 3 hours to the CreateDate and ModifyDate timestamps of two images.

`exiftool -AllDates+=1:30 -if '$make eq "Canon"' dir`

Shift the values of DateTimeOriginal, CreateDate and ModifyDate forward by 1 hour and 30 minutes for all Canon images in a directory. (The AllDates tag is provided as a shortcut for these three tags,

allowing them to be accessed via a single tag.)

```
exiftool -xmp:city=Kingston image1.jpg image2.nef
```

Write a tag to the XMP group of two images. (Without the `xmp:` this tag would get written to the IPTC group since `City` exists in both, and IPTC is preferred by default.)

```
exiftool -LightSource==Unknown (0) dst.tiff
```

Delete `LightSource` tag only if it is unknown with a value of 0.

```
exiftool -whitebalance==auto -WhiteBalance=tung dst.jpg
```

Set `WhiteBalance` to `Tungsten` only if it was previously `Auto`.

```
exiftool -comment== -comment='new comment' a.jpg
```

Write a new comment only if the image doesn't have one already.

```
exiftool -o %d%f.xmp dir
```

Create XMP meta information data files for all images in `dir`.

```
exiftool -o test.xmp -owner=Phil -title='XMP File'
```

Create an XMP data file only from tags defined on the command line.

```
exiftool '-ICC_Profile<=%d%f.icc' image.jpg
```

Write `ICC_Profile` to an image from a `.icc` file of the same name.

```
exiftool -hierarchicalkeywords='{keyword=one,children={keyword=B}}'
```

Write structured XMP information. See <<https://exiftool.org/struct.html>> for more details.

```
exiftool -trailer:all= image.jpg
```

Delete any trailer found after the end of image (EOI) in a JPEG file. A number of digital cameras store a large `PreviewImage` after the JPEG EOI, and the file size may be reduced significantly by deleting this trailer. See the JPEG Tags documentation for a list of recognized JPEG trailers.

COPYING EXAMPLES

These examples demonstrate the ability to copy tag values between files.

```
exiftool -tagsFromFile src.cr2 dst.jpg
```

Copy the values of all writable tags from `src.cr2` to `dst.jpg`, writing the information to same-named tags in the preferred groups.

```
exiftool -TagsFromFile src.jpg -all:all dst.jpg
```

Copy the values of all writable tags from `src.jpg` to `dst.jpg`, preserving the original tag groups.

```
exiftool -all= -tagsfromfile src.jpg -exif:all dst.jpg
```

Erase all meta information from `dst.jpg` image, then copy EXIF tags from `src.jpg`.

```
exiftool -exif:all= -tagsfromfile @ -all:all -unsafe bad.jpg
```

Rebuild all EXIF meta information from scratch in an image. This technique can be used in JPEG images to repair corrupted EXIF information which otherwise could not be written due to errors. The `Unsafe` tag is a shortcut for unsafe EXIF tags in JPEG images which are not normally copied. See the tag name documentation for more details about unsafe tags.

```
exiftool -Tagsfromfile a.jpg out.xmp
```

Copy meta information from `a.jpg` to an XMP data file. If the XMP data file `out.xmp` already exists, it will be updated with the new information. Otherwise the XMP data file will be created. Only metadata-only files may be created like this (files containing images may be edited but not created). See "WRITING EXAMPLES" above for another technique to generate XMP files.

```
exiftool -tagsFromFile a.jpg -XMP:All= -ThumbnailImage= -m b.jpg
```

Copy all meta information from `a.jpg` to `b.jpg`, deleting all XMP information and the thumbnail image from the destination.

```
exiftool -TagsFromFile src.jpg -title -author=Phil dst.jpg
```

Copy title from one image to another and set a new author name.

`exiftool -TagsFromFile a.jpg -ISO -TagsFromFile b.jpg -comment dst.jpg`
 Copy ISO from one image and Comment from another image to a destination image.

`exiftool -tagsfromfile src.jpg -exif:all --subifd:all dst.jpg`
 Copy only the EXIF information from one image to another, excluding SubIFD tags.

`exiftool '-FileModifyDate<DateTimeOriginal' dir`
 Use the original date from the meta information to set the same file's filesystem modification date for all images in a directory. (Note that `-TagsFromFile @` is assumed if no other `-TagsFromFile` is specified when redirecting information as in this example.)

`exiftool -TagsFromFile src.jpg '-xmp:all<all' dst.jpg`
 Copy all possible information from `src.jpg` and write in XMP format to `dst.jpg`.

`exiftool '-Description<${FileName;s\.[^\.]*$//}' dir`
 Set the image Description from the file name after removing the extension. This example uses the "Advanced formatting feature" to perform a substitution operation to remove the last dot and subsequent characters from the file name.

`exiftool -@ iptc2xmp.args -iptc:all= a.jpg`
 Translate IPTC information to XMP with appropriate tag name conversions, and delete the original IPTC information from an image. This example uses `iptc2xmp.args`, which is a file included with the ExifTool distribution that contains the required arguments to convert IPTC information to XMP format. Also included with the distribution are `xmp2iptc.args` (which performs the inverse conversion) and a few more `.args` files for other conversions between EXIF, IPTC and XMP.

`exiftool -tagsfromfile %d%f.CR2 -r -ext JPG dir`
 Recursively rewrite all JPG images in `dir` with information copied from the corresponding CR2 images in the same directories.

`exiftool '-keywords+<make' image.jpg`
 Add camera make to list of keywords.

`exiftool '-comment<ISO=$exif:iso Exposure=${shutterspeed}' dir`
 Set the Comment tag of all images in `dir` from the values of the EXIF:ISO and ShutterSpeed tags. The resulting comment will be in the form "ISO=100 Exposure=1/60".

`exiftool -TagsFromFile src.jpg -icc_profile dst.jpg`
 Copy ICC_Profile from one image to another.

`exiftool -TagsFromFile src.jpg -all:all dst.mie`
 Copy all meta information in its original form from a JPEG image to a MIE file. The MIE file will be created if it doesn't exist. This technique can be used to store the metadata of an image so it can be inserted back into the image (with the inverse command) later in a workflow.

`exiftool -o dst.mie -all:all src.jpg`
 This command performs exactly the same task as the command above, except that the `-o` option will not write to an output file that already exists.

`exiftool -b -jpgfromraw -w %d%f_%ue.jpg -execute -b -previewimage -w %d%f_%ue.jpg -execute -tagsfromfile @ -srcfile %d%f_%ue.jpg -overwrite_original -common_args --ext jpg DIR`
 [Advanced] Extract JpgFromRaw or PreviewImage from all but JPG files in DIR, saving them with file names like `image_EXT.jpg`, then add all meta information from the original files to the extracted images. Here, the command line is broken into three sections (separated by `-execute` options), and each is executed as if it were a separate command. The `-common_args` option causes the `--ext jpg DIR` arguments to be applied to all three commands, and the `-srcfile` option allows the extracted JPG image to be the source file for the third command (whereas the RAW files are the source files for the other two commands).

RENAMING EXAMPLES

By writing the `FileName` and `Directory` tags, files are renamed and/or moved to new directories. This can be particularly useful and powerful for organizing files by date when combined with the `-d` option.

New directories are created as necessary, but existing files will not be overwritten. The format codes %d, %f and %e may be used in the new file name to represent the directory, name and extension of the original file, and %c may be used to add a copy number if the file already exists (see the `-w` option for details). Note that if used within a date format string, an extra '%' must be added to pass these codes through the date/time parser. (And further note that in a Windows batch file, all '%' characters must also be escaped, so in this extreme case '%%%%%' is necessary to pass a simple '%f' through the two levels of parsing.) See <<https://exiftool.org/filename.html>> for additional documentation and examples.

```
exiftool -filename=new.jpg dir/old.jpg
```

Rename old.jpg to new.jpg in directory dir.

```
exiftool -directory=%e dir
```

Move all files from directory dir into directories named by the original file extensions.

```
exiftool '-Directory<DateTimeOriginal' -d %Y/%m/%d dir
```

Move all files in dir into a directory hierarchy based on year, month and day of DateTimeOriginal. eg) This command would move the file dir/image.jpg with a DateTimeOriginal of 2005:10:12 16:05:56 to 2005/10/12/image.jpg.

```
exiftool -o . '-Directory<DateTimeOriginal' -d %Y/%m/%d dir
```

Same effect as above except files are copied instead of moved.

```
exiftool '-filename<%f_{$model;}.%e' dir
```

Rename all files in dir by adding the camera model name to the file name. The semicolon after the tag name inside the braces causes characters which are invalid in Windows file names to be deleted from the tag value (see the "Advanced formatting feature" for an explanation).

```
exiftool '-FileName<CreateDate' -d %Y%m%d_%H%M%S%-c.%e dir
```

Rename all images in dir according to the CreateDate date and time, adding a copy number with leading '-' if the file already exists (%-c), and preserving the original file extension (%e). Note the extra '%' necessary to escape the filename codes (%c and %e) in the date format string.

```
exiftool -r '-FileName<CreateDate' -d %Y-%m-%d/%H%M_%.%f.%e dir
```

Both the directory and the filename may be changed together via the FileName tag if the new FileName contains a '/'. The example above recursively renames all images in a directory by adding a CreateDate timestamp to the start of the filename, then moves them into new directories named by date.

```
exiftool '-FileName<${CreateDate}_$filename.jpg' -d %Y%m%d -ext jpg .
```

Set the filename of all JPG images in the current directory from the CreateDate and FileNumber tags, in the form "20060507_118-1861.jpg".

GEOTAGGING EXAMPLES

ExifTool implements geotagging via 3 special tags: Geotag (which for convenience is also implemented as an exiftool option), Geosync and Geotime. The examples below highlight some geotagging features. See <<https://exiftool.org/geotag.html>> for additional documentation.

```
exiftool -geotag track.log a.jpg
```

Geotag an image (a.jpg) from position information in a GPS track log (track.log). Since the Geotime tag is not specified, the value of DateTimeOriginal is used for geotagging. Local system time is assumed unless DateTimeOriginal contains a timezone.

```
exiftool -geotag t.log -geotime='2009:04:02 13:41:12-05:00' a.jpg
```

Geotag an image with the GPS position for a specific time.

```
exiftool -geotag log.gpx '-xmp:geotime<createdate' dir
```

Geotag all images in directory dir with XMP tags instead of EXIF tags, based on the image CreateDate.

```
exiftool -geotag a.log -geosync=-20 dir
```

Geotag images in directory dir, accounting for image timestamps which were 20 seconds ahead of GPS.

```
exiftool -geotag a.log -geosync=1.jpg -geosync=2.jpg dir
    Geotag images using time synchronization from two previously geotagged images (1.jpg and 2.jpg),
    synchronizing the image and GPS times using a linear time drift correction.
```

```
exiftool -geotag a.log '-geotime<${createdate}+01:00' dir
    Geotag images in dir using CreateDate with the specified timezone. If CreateDate already
    contained a timezone, then the timezone specified on the command line is ignored.
```

```
exiftool -geotag= a.jpg
    Delete GPS tags which may have been added by the geotag feature. Note that this does not remove
    all GPS tags — to do this instead use -gps:all=.
```

```
exiftool -xmp:geotag= a.jpg
    Delete XMP GPS tags which were added by the geotag feature.
```

```
exiftool -xmp:geotag=track.log a.jpg
    Geotag an image with XMP tags, using the time from DateTimeOriginal.
```

```
exiftool -geotag a.log -geotag b.log -r dir
    Combine multiple track logs and geotag an entire directory tree of images.
```

```
exiftool -geotag 'tracks/*.log' -r dir
    Read all track logs from the tracks directory.
```

```
exiftool -p gpx.fmt -d %Y-%m-%dT%H:%M:%SZ dir > out.gpx
    Generate a GPX track log from all images in directory dir. This example uses the gpx.fmt file
    included in the full ExifTool distribution package and assumes that the images in dir have all been
    previously geotagged.
```

PIPING EXAMPLES

```
cat a.jpg | exiftool -
    Extract information from stdin.
```

```
exiftool image.jpg -thumbnailimage -b | exiftool -
    Extract information from an embedded thumbnail image.
```

```
cat a.jpg | exiftool -iptc:keywords+=fantastic - > b.jpg
    Add an IPTC keyword in a pipeline, saving output to a new file.
```

```
curl -s http://a.domain.com/bigfile.jpg | exiftool -fast -
    Extract information from an image over the internet using the cURL utility. The -fast option
    prevents exiftool from scanning for trailer information, so only the meta information header is
    transferred.
```

```
exiftool a.jpg -thumbnailimage -b | exiftool -comment=wow - | exiftool a.jpg -thumbnailimage'<==-'
    Add a comment to an embedded thumbnail image. (Why anyone would want to do this I don't know,
    but I've included this as an example to illustrate the flexibility of ExifTool.)
```

EXIT STATUS

The `exiftool` application exits with a status of 0 on success, or 1 if an error occurred, or 2 if all files failed the `-if` condition (for any of the commands if `-execute` was used).

AUTHOR

Copyright 2003–2020, Phil Harvey

This is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

SEE ALSO

Image::ExifTool (3pm), *Image::ExifTool::TagNames* (3pm), *Image::ExifTool::Shortcuts* (3pm),
Image::ExifTool::Shift.pl

NAME

exrc – vi/vim initialization file

SYNOPSIS

edrc/etc/exrc

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **vi**(1) and **vim**(1) command.

FILEFORMAT

The fileformat is a plain **ASCII** file without the capability of defining comments.

OPTIONS

-

EXAMPLES

```
set autoindent autoprint nomesg noslowopen noterse shiftwidth=4
set report=2 tabstop=4 wrapmargin=0 nu showmatch showmode
set noerrorbells
set cmdheight=2
map ; :
map g :%
map Q :q!
cmap mono set t_Co=0
cmap color set t_Co=8
```

SEE ALSO

edrcintro(1), **vi**(1), **vim**(1)

NOTES

-

BUGS

-

AUTHOR

exrc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fcreate – create a file if it does not exist

SYNOPSIS

edrc/lib/fcreate *file* [*text*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

create a file if it does not exist. Optionally text can be written to the file created.

The intention of this command is to use it to create lock files within shell scripts. Or to atomically create a file if it does not exist.

The created *file* has the file permissions 0600.

OPTIONS

file name of the file to be created.

text text to be written to the file specified in the **file** option.

EXIT STATUS

0 specified file did not exist before and could be created.

1 specified file already exists.

2 error occurred during file creation.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **open**(3)

NOTES

-

BUGS

-

AUTHOR

fcreate was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fields2swvi – filter to format data stream with named swvi fields to swvi sequence

SYNOPSIS

edrc/bin/fields2swvi [**-h**]

stream | fields2swvi

fields2swvi < *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **fields2swvi** command is a filter that brings an unordered software inventory 'FIELD = VALUE' data stream into the correct sequence as understood by **swvi**(1).

This command is internally used by **apply2sw_inventory**(1).

OPTIONS

-h usage message.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

5 the command has been aborted pressing *Ctrl+C*.

FILES

-

EXAMPLES**1) Example input/output:**

The input sequence is not important, but only known *FIELDS* are printed, unknown *FIELDS* are ignored.

Input:

```
cat <<EOM | fields2swvi
VERSION      = 1.5.02
SOFTWARE     = EDRC - Enterprise Disaster Recovery Console
INST_USER    = root
INST_GROUP   = root
RUN_USER     =
RUN_GROUP    =
CFG_DIR      = etc
VAR_DIR      = var
LICENSE      = GNU GPL
LIC_MGMT     =
WEB          =
DOC          = ~edrc/doc/edrc_manpages-1.5.02.pdf
INST_SRC     = $Pkgfile
PRODUCT      = System
INSTANCE     =
COMMENT      = disaster recovery and system administration
EOM
```

Output:

```
EDRC - Enterprise Disaster Recovery Console
1.5.02
root
root

etc
var
GNU GPL

~edrc/doc/edrc_manpages-1.5.02.pdf
$Pkgfile
System

disaster recovery and system administration
y
y
```

SEE ALSO

edrcintro(1), apply2sw_inventory(1) sw_inventory(4) sw_report(1), swvi(1),

NOTES

-

BUGS

-

AUTHOR

fields2swvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

filedist – distribute file(s) to a list of remote systems

SYNOPSIS

edrc/bin/filedist [**-h**]

filedist [**-f**] [**-t** *host_1,host_2,host_n*] *file* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

Distribute given file(s) to an interactively queried list of hosts.

The file(s) are distributed to the identical path as they are located on the source system.

The interactive query of hosts suggests as default the hosts as defined in the **DIST_HOSTLIST** setting in the **rcmd.cfg** file. Furthermore it is possible to enter hostgroups (*@HOSTGROUP*), as known from the **hostlist** command. It is possible to specify multiple hostgroups or to mix hostgroups with hostnames. When using hostgroups the input default is expanded and the target prompt is repeated to give the user the opportunity to verify the target list and to correct it if needed.

The host where logged on is excluded from distribution even if it is on the target list.

It is possible to block files from being distributed from or to certain hosts. The blocking rules are specified in the **etc/filedist.block** file. The intention of blocking files is to avoid miss-configurations etc. by distributing files to/from certain hosts by error.

OPTIONS

-h usage message.

-f force distribution of files blocked as defined in **etc/filedist.block**.

-t *hostlist* A comma separated list of target hosts. If the hostlist is not specified, an interactive query asks for it.

See also **resolve_targetlist(3)** to see how to efficiently create a hostlist in scripts.

file ... space separated list of files to be distributed to one or more target systems. The file(s) are distributed to the identical path as it resides on the source system. Directories and symbolic links are not accepted as source in the *file* option. When the file intended to be distributed is a directory or symbolic link on the target system it will not be distributed to that system and a error message is displayed. Also if a *file* is located in a directory on the source system but the directory does not exist on the target system, it is not distributed and an error message will inform

about this.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 2** operating system is not supported, yet. See **osid**(3) if you get this error.
- 4** usage printed.
- 5** the distribution has been aborted using *Ctrl+C*.
- 6** the configuration file **filedist.cfg** does not exist.
- 7** the blockfile **filedist.block** does not exist.
- 11** temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

etc/filedist.cfg

configuration file of **filedist**, see **filedist.cfg**(4) for more information.

etc/filedist.block

configuration file of **filedist** to prevent (block) files from being distributed from/to certain targets, see **filedist.block**(4) for more information.

var/log/filedist.log

default logfile of **filedist**.

EXAMPLES

1) normal target list

Distribute files to a list of targets. In this example the file **/etc/protocol** does not exist on the source node. The file **/etc/init.d** is a directory. On the system dcdbsi63 the file **/etc/telnetd** is a symbolic link, therefore the file is not distributed from the source node to this host.

```
[ /etc ]
[ root@dcdbsi74 ][-sh]: filedist hosts services protocol telnetd init.d
filedist - copy files to other systems, by Chr. Walther
```



```
Target hosts: [ dcdbsi63 dcdbsi64 dcdbsi73 dcdbsi74 ]:
```

```
distribute files ...
  host dcdbsi63 ...
    file /etc/hosts ... done
    file /etc/services ... done
    file protocol ...(not found)... skip
    file /etc/telnetd ...(link)... skip
    file init.d ...(not found)... skip
  done
  host dcdbsi64 ...
    file /etc/hosts ... done
    file /etc/services ... done
    file protocol ...(not found)... skip
    file /etc/telnetd ...(link)... skip
    file init.d ...(not found)... skip
  done
  host dcdbsi73 ...
    file /etc/hosts ... done
    file /etc/services ...(target is link)... fail
    file protocol ...(not found)... skip
    file /etc/telnetd ...(link)... skip
    file init.d ...(not found)... skip
  done
  host dcdbsi74 ...(localhost)... skip
done
```

2) target list using hostgroups

Distribute files to a list of targets using hostgroups. In this example the file **/etc/hosts** is blocked to be distributed to the host dcdbsi50. The host dcdbsi99 is not up.

```
[ / ]
[ root@dcdbsi70 ][-sh]: filedist /hosts/hosts
filedist - copy files to other systems, by Chr. Walther
```

```
Target hosts: [ dcdbsi60 dcdbsi70 ]: @ALL dcdbsi50 dcdbsi99
Target hosts: [ dcdbsi05 dcdbsi10 dcdbsi20 ... dcdbsi73 dcdbsi74 dcdbsi99
```

```
distribute files ...
  host dcdbsi05 ...
    file /etc/hosts ... done
  done
  host dcdbsi10 ...
    file /etc/hosts ... done
  done
  :
  :
  host dcdbsi33 ...
    file /etc/hosts ... done
  done
  host dcdbsi50 ...
    file /etc/hosts ...(blocked)... skip
  done
```

```

host dcdbsi51 ...
    file /etc/hosts ... done
done
:
:
host dcdbsi64 ...
    file /etc/hosts ... done
done
host dcdbsi70 ...(localhost)... skip
host dcdbsi71 ...
    file /etc/hosts ... done
done
:
:
host dcdbsi74 ...
    file /etc/hosts ... done
done
host dcdbsi99 ...(not up)... skip
done

```

3) use filedist in scripts (host only targetlist)

```

echo "host-001 host-002 \n" | \
filedist /etc/hosts

```

4) use filedist in scripts (hostgroups in targetlist)

```

echo "`resolve_targetlist -t host-001 host-002 @BE` \n" | \
filedist /etc/hosts

```

or:

```

echo "host-001 host-002 @BE \n\n" | \
filedist /etc/hosts

```

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **hostaliases(3)**, **remote_shell(3)**, **remote_copy(3)**, **filedist.cfg(4)**, **filedist.block(4)**, **resolve_targetlist(3)**, **hostlist(3)**

NOTES

When distributing files as user root from the source system, the user as defined in **DIST_USER** in the configuration file **etc/filedist.cfg** is used to connect to the target systems. On the target systems this user has to be allowed to switch to root in the related security files, see **edrcsetup(1m)** for an explanation of the required security settings.

If **filedist** is used as another user as root, the own user is used for the distribution.

BUGS

-

AUTHOR

filedist was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

filedist.cfg – configuration file for filedist

SYNOPSIS

edrc/etc/filedist.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **filedist** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **filedist**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

DIST_HOSTLIST

Space separated list of hosts. To this hosts the files specified on the command line will be distributed.

Example: DIST_HOSTLIST="acme045 acme014"

Default: –

DIST_USER

User used to distribute the files if the user invoking **filedist** is root. Other users use the own user for the distribution.

Example: DIST_USER=edrc

Default: DIST_USER=edrc

DIST_MODE

Comma separated list of modes used to distribute the files in **SCRIPTS_BASEDIR**. The supported modes are: *rtools* which result in the use of **rcp** and **rsh** or *OpenSSH* which results in the use of **scp** and **ssh** for distribution. If a comma separated list is provided, the connection initiation is made in the sequence specified. A pseudo distribution mode is *default* which results in the use of the **CONNECTION_MODE** specified in the configuration files **remote_shell.cfg** and **remote_copy.cfg**. It is not allowed to specify *default* as part of a comma separated list.

Example: DIST_MODE=rtools,OpenSSH

Default: DIST_MODE=rtools

REMOTE_CMD_PATH

Path on the remote systems to the **cpio** and **gzip** commands.

Example: REMOTE_CMD_PATH=/usr/contrib/bin:/usr/local/bin:/opt/edrc/bin

Default: REMOTE_CMD_PATH=

SUMMARY

If set to *True* at the end of the file distribution execution a state summary is printed. If set to *False* the summary is not printed.

Example: SUMMARY=True

Default: SUMMARY=False

SEE ALSO

edrcintro(1), **filedist**(1), **filedist.block**(4), **hostlist**(3)

NOTES

-

BUGS

-

AUTHOR

filedist.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

filedist.block – configuration file for filedist

SYNOPSIS

edrc/etc/filedist.block

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file to block files from distribution using the **filedist** command.

FILEFORMAT

The fileformat is a list of rules that have the format

mode:source_host:destination_host:file

The mode field specifies if a rule is to **allow** or **block** a file being distributed from a source- to a destination host.

The fields *source_host*, *destination_host* and *file* are regular expressions.

The rules are processed in the sequence listed. If a rule matches, the remaining rules are not processed any more.

The last rule that is processed by **filedist** is always

```
block:.*:.*:.*
```

if no previous rule matched. Therefore if you like to allow all distributions except the defined blocked files, the rule

```
allow:.*:.*:.*
```

has to be added as the very last rule in the **etc/filedist.block** file.

Regular expression hints:

```
.*
```

Everything. This can be used to specify all hosts or all files.

dcdbsi5[1-3]

Range. This example defines the hosts dcdbsi51, dcdbsi52 and dcdbsi53.

See also **EXAMPLES** section.

OPTIONS

- mode** mode to define the rules what files can be distributed:
- allow** if the rule definition of *source_host*, *destination_host* and *file* matches, the file is allowed to be distributed.
 - block** if the rule definition of *source_host*, *destination_host* and *file* matches, the file is blocked from distribution.
- source_host* Regular expression of the source host. The source host to be dealt with is as it is resolved by the **hostname(1)** command.
- destination_host* Regular expression of the source host. The destination host to be dealt with is as it is specified in the interactive "Target hosts:" query of the **filedist** command. When specifying the *destination_host* take also into account, that the user could specify aliases for distribution targets or use a full qualified name. In most cases it wouldn't be needed to specify all aliases due to the provided default targets that should lead the user into the right direction.
- file* Regular expression of the file to be distributed.

EXAMPLES

1) block rule example

This definition helps, that only files in the directories **/opt/edrc**, the installation directory of WA2L/edrc, and **/tmp** are allowed to be distributed from and to dcdbsi50.

This example has the background, that the host dcdbsi50 is a Sun Solaris system and all other hosts are HP-UX systems. To avoid miss-configuration of the systems by distributing operating system files that might have a different format on either one of the two operating systems, only a very reduced set of files is allowed to be distributed.

The last rule in this example ensures, that between all other systems all files are allowed to be distributed.

```
#
# filedist.block - filedist configuration file to block
#                  (prevent) files from distribution
#
# [00] 11.08.2003 CWa   Initial Version
#
#
# Fileformat:
#
```



```

# <mode>:<source_host>:<destination_host>:<file>
#
# Fields:
#     <mode>          = allow or block
#     <source_host>, <destination_host> and <file>
#                     are regular expressions
#
# Hints:
#     all hosts      = .*
#     all files      = .*
#     range          = dcdbsi5[1-2]
#
#
# RULES:
#
# The rules are processed in the sequence listed.
# If a rule matches, the remaining rules are not
# processed any more.
#
#
# allow only /opt/edrc/* and /tmp/* files to be
# distributed from and to dcdbsi50.
#
allow:.*:.*:/opt/edrc/.*
allow:.*:.*:/tmp/.*
block:.*:dcdbsi50:.*
block:dcdbsi50:.*:.*
#
# last rule, allow all distributions, when no
# previous rule matched.
#
allow:.*:.*:.*

```

SEE ALSO

edrcintro(1), **filedist(1)**, **hostlist(3)**, **regexintro(4)**

NOTES

The *destination_host* regular expression is checked against the interactive user input. If the user specifies a host alias, the *destination_host* regular expression might not match and the file is distributed even if it was intended to be blocked.

BUGS

-

AUTHOR

filedist.block was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

filegrep – grep underlying recovery scripts for a pattern

SYNOPSIS

edrc/bin/filegrep *pattern*

AVAILABILITY

WA2L/edrc

DESCRIPTION

grep for a pattern in all underlying ASCII files. Files located in a **.sav** directory or files saved with the **sav(1)** command are excluded from the search.

OPTIONS

pattern regular expression pattern to be searched for in underlying file. See **regexintro(4)** for more information about regular expressions.

ENVIRONMENT

-

EXIT STATUS

4 no pattern defined, usage displayed.
X exit code of **grep(1)**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **grep(1)**, **regexintro(4)**, **scriptgrep(1)**

NOTES

-

BUGS

-

AUTHOR

filegrep was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME**SYNOPSIS**

edrc/bin/filelink [**-h**]

filelink -m mode -d directory_1 -d directory_2

AVAILABILITY

WA2L/edrc

DESCRIPTION

create hardlinks between identical files in two directory trees.

The hardlinks are created between files having the identical name and content in the two directories specified.

The main intention of the command is to simplify configuration file management when using **sysconfig(1m)** and having different configurations having some identical files.

OPTIONS

-h usage message.

-m mode mode of the program execution:

print

print expected changes in *directory_2*

execute

execute changes in *directory_2*

-d directory_1
source directory.

-d directory_2
directory where the links are created.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|--------------------------------------|
| 0 | no error. |
| 2 | operating system not supported. |
| 3 | cannot write to <i>directory_1</i> . |
| 4 | usage message displayed. |
| 5 | command aborted. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ln(1)**, **sysconfig(1)**

NOTES

-

BUGS

-

AUTHOR

filelink was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

filesize – print size of a file

SYNOPSIS

edrc/lib/filesize [**-h**]

filesize [**-b** | **-k** | **-m** | **-g**] **-f** *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the size of a given file.

The file size is returned as an integer number of the selected unit option. If the file size is not integer the returned file size is rounded up by one. Therefore if the real file size is 123.01 kBytes, 124 would be returned.

If a more precise file size is needed for units other than Bytes (**-b**) the size has to be queried in Bytes first and then the related calculation has to be done separately:

```
filesize -f /data/example.dmp | \
awk '{printf("%3.2f\n", $1/1024)}'
```

OPTIONS

- h** print usage message.
- b** return the file size in Bytes, this is also the default if no unit option is specified.
- k** return the file size in kilo bytes (1 kByte = 1024 Byte).
- m** return the file size in mega bytes (1 MByte = 1024 kByte).
- g** return the file size in giga bytes (1 GByte = 1024 MByte).
- f** *file* path of the file from which the size should be returned.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---|
| 0 | no error. |
| 1 | given file does not exist. |
| 2 | operating system does not exist. See osid (3) if you get this error. |
| 4 | usage message printed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **filewatch**(3), **freespace**(3), **bdf**(1), **df**(1)

NOTES

-

BUGS

-

AUTHOR

filesize was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

filewatch – print file size progress of a command

SYNOPSIS

edrc/lib/filewatch [**-h**]

filewatch *file* (*command* | *PID*)

AVAILABILITY

WA2L/edrc

DESCRIPTION

watch the file size progress of a given running command.

OPTIONS

-h print usage message.

file file to watch the file size growth.

command command that produces the *file*.

PID process ID of the command that produces the *file*.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 given file does not exist.

2 operating system does not exist. See **osid(3)** if you get this error.

4 usage message printed.

5 **filewatch** aborted.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **filesize(3)**, **freespace(3)**, **pslist(3)**

NOTES

-

BUGS

-

AUTHOR

filewatch was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fit2width – fit a stream to a with

SYNOPSIS

edrc/lib/fit2width [**-h** | **-m**] [*width*]

edrc/lib/fit [**-h** | **-m**] [*width*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

fit a data stream to a specified width.

This command can be used together with **tty_columns(3)** to fit a stream to the current width of a terminal (tty).

OPTIONS

-h print usage message.

-m mark lines that are longer as the specified *width* with '>>' at the end of the line.
If this option is not specified, the lines are shortened to the given *width* without marking it.

width number of characters to be printed. If this option is not set, the default width is resolved by **tty_columns(3)** if it is a tty. If it is not a tty the contents of the **\$COLUMNS** environment variable is used as a default.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message printed.

FILES

-

EXAMPLES**1) fit the /etc/services file to the screen width using tty_columns**

```
[ / ]
[ root@acme007 ][-sh]: cat /etc/services | \
                        fit2width -m `tty_columns`

#
# /etc/services: - Network services, Internet Style
#
# Each line describes one service, and is of the for >>
#
# service-name  port/protocol  [aliases ...]  [# co >>

tcpmux          1/tcp                # TC >>
tcpmux          1/udp                # TC >>
rje             5/tcp                # Re >>
:
:
```

2) fit the /etc/services file to the screen width using default width

```
[ / ]
[ root@acme007 ][-sh]: cat /etc/services | \
                        fit2width -m

#
# /etc/services: - Network services, Internet Style
#
# Each line describes one service, and is of the for >>
#
# service-name  port/protocol  [aliases ...]  [# co >>

tcpmux          1/tcp                # TC >>
tcpmux          1/udp                # TC >>
rje             5/tcp                # Re >>
:
:
```

3) fit the /etc/services file to a width of 55 characters

```
[ / ]
[ root@acme007 ][-sh]: cat /etc/services | \
                        fit2width -m 55

#
# /etc/services: - Network services, Internet Style
#
```

```
# Each line describes one service, and is of the form >>
#
# service-name port/protocol [aliases ...] [# comment >>

tcpmux          1/tcp          # TC >>
tcpmux          1/udp          # TC >>
rje             5/tcp          # Re >>
:
:
```

SEE ALSO

edrcintro(1), **tty_columns(3)**

NOTES

-

BUGS

-

AUTHOR

fit2width was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fnmatch – wildcard pattern match of a stream

SYNOPSIS

edrc/lib/fnmatch [-h | -V]

fnmatch [-n] [-m *num*] [-i] [-v] [-e] *pattern* < *file*

cat *file* | fnmatch [-n] [-m *num*] [-i] [-v] [-e] *pattern*

AVAILABILITY

WA2L/edrc

DESCRIPTION

match a stream by a wildcard pattern expression.

Wildcard expressions are different to regular expressions and are known for example from the **ls**(1) command.

However, in some applications wildcard patterns are more intuitive to use then regular expressions.

OPTIONS

-h usage message.

-V print command version.

-n prefix each line of output with the 1-based line number within its input file.

-m *num* stop reading a file after *num* matching lines.

-i case insensitive match.

-e extended expression as known in **ksh**:

The extended format is as follows, with *pattern-list* being a `'|'` separated list of *patterns*.

`'?(pattern-list)'`

The *pattern* matches if zero or one occurrences of any of the *patterns* in the *pattern-list* match the input stream.

`'*(pattern-list)'`

The *pattern* matches if zero or more occurrences of any of the *patterns* in the *pattern-list* match the input stream.

'+(*pattern-list*)'

The *pattern* matches if one or more occurrences of any of the *patterns* in the *pattern-list* match the input stream.

'@(*pattern-list*)'

The *pattern* matches if exactly one occurrence of any of the *patterns* in the *pattern-list* match the input stream.

'!(*pattern-list*)'

The *pattern* matches if the input stream cannot be matched

-v invert the sense of matching, to select non-matching lines.

pattern wildcard expression to match the stream:

***** Matches any string, including the null string.

? Matches any single character.

[?] Matches the question mark (?) character.

[...] Matches any one of the characters enclosed. A pair of characters separated by - matches any character lexically between the pair.

[!...] Matches any character except the enclosed ones.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|------------------|
| 0 | match |
| 1 | no match. |
| 4 | usage printed. |
| 5 | version printed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **egrep(1)**, **fgrep(1)**, **grep(1)**, **ksh(1)**, **regexintro(4)**

NOTES

-

BUGS

-

AUTHOR

fnmatch was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fmatch – return values for a given key from a data file

SYNOPSIS

edrc/lib/fmatch [**-h**]

fmatch -k "*key_regex*" [**-f** *data_file*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

prints the values of the given *key_regex* from the file csv database specified on command line using the **-f** option or in the **\$FMATCH_DATA_FILE** environment variable.

The structure of the *data_file* is:

key;values_list;

where empty lines and lines starting with a **#** are considered as comments. The backslash can be used to define rows spanning multiple lines in the *data_file*.

The **values_list** is a white space separated list of values for a certain key and are returned as a uniquely sorted list to **stdout**.

OPTIONS

-h usage message.

-k "*key_regex*"
key as regular expression as known from the **egrep**(1) command for selecting the related values in the *data_file*.

-f *data_file*
file to select the values from. If the environment variable **\$FMATCH_DATA_FILE** is set this option is optional. The *data_file* must have the following format:

key;values;

Empty lines and lines starting with a **#** are considered as comments. The backslash can be used to define rows spanning multiple lines in the *data_file*.

Example *data_file* :

#

```
# etc/hostlist.dat - hostlist csv database file
#
# [00] 12.08.2008 CWa    Initial Version
#
#
# Format:
#
#   key ::= <Customer>--<SERVER_ENVIRONMENT>
#
#   <key>;<value_list>;
#
ACME--INFRASTRUCTURE;    acme001 acme002;

LooneyTunes--TEST;      ltunes001 ltunes005 ltunes008;
LooneyTunes--PRODUCTION;ltunes002 ltunes003 ltunes004 \
                        ltunes006 ltunes007;

BOULDER--DEVELOPMENT;   boulder000 boulder001 boulder002;
BOULDER--TEST;          boulder100 boulder101 boulder102;
BOULDER--PRODUCTION;    boulder201 boulder201 boulder202 \
                        boulder203;
```

ENVIRONMENT**\$FMATCH_DATA_FILE**

data_file as specified in the **-f** option. When this environment variable is set, the **-f** option is optional.

EXIT STATUS

| | |
|----------|-----------------------|
| 0 | no error. |
| 4 | usage message listed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **egrep**(4), **regexintro**(4), **ypmatch**(1)

NOTES

-

BUGS

-

AUTHOR

fmatch was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

freespace – print freespace of a directory

SYNOPSIS

edrc/lib/freespace [**-h**]

freespace [**-b** | **-k** | **-m** | **-g**] **-d** *directory*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the available free space of the filesystem where the given directory resides on.

The free space is returned as an integer number of the selected unit option.

If a more precise free space size is needed for units other than Bytes (**-b**) the size has to be queried in Bytes first and then the related calculation has to be done separately:

```
freespace -d /data | \
awk '{printf("%3.2f\n", $1/1024)}'
```

OPTIONS

- h** print usage message.
- b** return the free space in Bytes, this is also the default if no unit option is specified.
- k** return the free space in kilo bytes (1 kByte = 1024 Byte).
- m** return the free space in mega bytes (1 MByte = 1024 kByte).
- g** return the free space in giga bytes (1 GByte = 1024 MByte).
- d** *directory*
path of the directory from where the free space should be returned.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---|
| 0 | no error. |
| 1 | given directory does not exist. |
| 2 | operating system does not exist. See osid (3) if you get this error. |
| 4 | usage message printed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **filesize**(3), **bdf**(1), **df**(1)

NOTES

-

BUGS

-

AUTHOR

freespace was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fssum – summarize and group filesystems by product

SYNOPSIS

edrc/bin/fssum [**-h**]

fssum [**-c** *cfg_file*] [**-d** [**-v**]]

fssum [**-c** *cfg_file*] [**-t** [**-v**]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

summarize filesystems based on definitions in a configuration file.

This command is used to create a filesystem summary grouped by product.

OPTIONS

-h usage message.

-c *config_file*

configuration file containing the grouping description, purpose and selection regular expressions. The default config file is **edrc/etc/fssum.cfg** when this option is not used.

-d CSV output to be used for further processing. If this option is not specified a structured ASCII report is printed.

-t print the CSV header as the first row of the CSV output.

-v "verbose" output in the CSV report, that prints the row number of the config file that matched for a certain file system.

ENVIRONMENT

-

EXIT STATUS

0 no error.

- 3** configuration file does not exist.
- 4** usage listed.
- 5** command aborted, e.g. by pressing <Ctrl+C>.

FILES

edrc/etc/fssum.cfg
configuration file for **fssum**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **egrep(1)**, **fssum.cfg(4)**, **regextintro(4)**

NOTES

-

BUGS

-

AUTHOR

fssum was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

fssum.cfg – configuration file for fssum

SYNOPSIS

edrc/etc/fssum.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration for the **fssum** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

type : system : group : title : purpose : storage : selection_regex :

Rows starting with a **#** are considered as comments.

For each file system all definitions are processed in the sequence as listed in the configuration file. A special case is when the *group* is set to '**EXCLUDE**' (see *group* description below).

OPTIONS

type system type:

h hostname

p cluster package name

system system- or cluster package name. This setting can be a regular expression.

If *localhost* is specified here, the *selection_regex* will be resolved on all hosts (system type = **h**), but in the output the real hostname as resolved by the **hostname(1)** command appears instead of *localhost*.

group general grouping selector. This selector can be used to group for example applications that are distributed over multiple nodes in the **fssum** report.

If the special group '**EXCLUDE**' is set, the file system as selected by the *selection_regex* regular expression is excluded completely from the report. To have the desired effect, the '**EXCLUDE**' entry has to be placed before the other definitions in the **fssum.cfg** file.

title description title of the file system selection. The ASCII report is grouped by the combination *system, title*. Therefore a normal **fssum.cfg** file will have multiple identical *title* definitions normally.

purpose purpose description text of a certain file system.

storage storage description, where a certain file system resides.

selection_regex
file system or mount point selection regular expressions.

The input string on which the **selection_regex** takes effect is the "Filesystem" and "Mounted" columns of the **xbdf** output. The two columns are separated by four spaces.

Example:

```
/dev/vg00/lvol0    /
/dev/vg00/lvol1    /var
/dev/sda1          /boot
tmpfs              /dev/shm
```

EXAMPLES

-

SEE ALSO

edrcintro(1), **fssum**(4), **regexintro**(3), **xbdf**(1m)

NOTES

-

BUGS

-

AUTHOR

fssum.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ftps – secure FTP to connect to the server using SSL

SYNOPSIS

edrc/bin/ftps [**-h**]

ftps [**-script** *script-file* | *host* [*port*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Secure FTP is an FTP program that allows for a secure connection to be made to an FTP server via a Secure Sockets Layer (SSL). This version of Secure FTP supports both implicit and explicit SSL connections.

With no port specified, an implicit SSL connection is attempted on port 990 (the default for implicit SSL connections). If a connection cannot be made implicitly, an explicit SSL connection is attempted on port 21 (the default for explicit SSL connections). These default ports can be changed by defining the properties:

```
default.port.ssl.implicit and default.port.ssl.explicit.
```

For example, you would set the default implicit port to 12345 by saying:

```
java -jar -Ddefault.port.ssl.implicit=12345 secureftp2.jar
```

Scripting Secure FTP:

Secure FTP supports simple scripting. A script is nothing more than a line-by-line representation of what would be entered at the command line. A sample script might look like:

```
# Sample Secure FTP script
open ftp.glub.com
user demo <pass>
lcd /tmp
mkdir test
lcd test
cd /pub
mget *
quit
```

To call this script, from the shell command line you would pass in the arguments:

```
-script <path to script>/<file>
```

Additional Help:

To get additional help, from within the program you can type `help` to get a listing of all available commands.

OPTIONS

-h usage message.

-script *script-file*
 start a ftp script.

host host name to connect to.

port port to connect with.

ENVIRONMENT

-

EXIT STATUS

-

FILES

etc/ftps.cfg
 configuration file of **ftps**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **java_wrapper(3)**, **lib/java/secureftp-*/SECUREFTP.README**,
lib/java/secureftp-*/SECUREFTP.LICENSE, <http://www.glub.com/products/secureftp>

NOTES

Parts of this manpage were extracted from the documentation of the **Glub Tech: Secure FTP README** file. See <http://www.glub.com/products/secureftp/> for more information.

BUGS

-

AUTHOR

ftps was developed by **Glub Tech** (<http://www.glub.com>) and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2012 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

gawk – pattern scanning and processing language

SYNOPSIS

gawk [POSIX or GNU style options] **-f** *program-file* [**--**] file ...

gawk [POSIX or GNU style options] [**--**] *program-text* file ...

pgawk [POSIX or GNU style options] **-f** *program-file* [**--**] file ...

pgawk [POSIX or GNU style options] [**--**] *program-text* file ...

DESCRIPTION

Gawk is the GNU Project's implementation of the AWK programming language. It conforms to the definition of the language in the POSIX 1003.1 Standard. This version in turn is based on the description in *The AWK Programming Language*, by Aho, Kernighan, and Weinberger, with the additional features found in the System V Release 4 version of UNIX *awk*. *Gawk* also provides more recent Bell Laboratories *awk* extensions, and a number of GNU-specific extensions.

Pgawk is the profiling version of *gawk*. It is identical in every way to *gawk*, except that programs run more slowly, and it automatically produces an execution profile in the file **awkprof.out** when done. See the **--profile** option, below.

The command line consists of options to *gawk* itself, the AWK program text (if not supplied via the **-f** or **--file** options), and values to be made available in the **ARGC** and **ARGV** pre-defined AWK variables.

OPTION FORMAT

Gawk options may be either traditional POSIX one letter options, or GNU-style long options. POSIX options start with a single “-”, while long options start with “--”. Long options are provided for both GNU-specific features and for POSIX-mandated features.

Following the POSIX standard, *gawk*-specific options are supplied via arguments to the **-W** option. Multiple **-W** options may be supplied. Each **-W** option has a corresponding long option, as detailed below. Arguments to long options are either joined with the option by an = sign, with no intervening spaces, or they may be provided in the next command line argument. Long options may be abbreviated, as long as the abbreviation remains unique.

OPTIONS

Gawk accepts the following options, listed by frequency.

-F *fs*

--field-separator *fs*

Use *fs* for the input field separator (the value of the **FS** predefined variable).

-v *var=val*

--assign *var=val*

Assign the value *val* to the variable *var*, before execution of the program begins. Such variable values are available to the **BEGIN** block of an AWK program.

-f *program-file*

--file *program-file*

Read the AWK program source from the file *program-file*, instead of from the first command line argument. Multiple **-f** (or **--file**) options may be used.

-mf *NNN*

-mr *NNN*

Set various memory limits to the value *NNN*. The **f** flag sets the maximum number of fields, and the **r** flag sets the maximum record size. These two flags and the **-m** option are from an earlier version of the Bell Laboratories research version of UNIX *awk*. They are ignored by *gawk*, since *gawk* has no pre-defined limits.

-W **compat**

-W traditional

--compat

--traditional

Run in *compatibility* mode. In compatibility mode, *gawk* behaves identically to UNIX *awk*; none of the GNU-specific extensions are recognized. The use of **--traditional** is preferred over the other forms of this option. See **GNU EXTENSIONS**, below, for more information.

-W copyleft

-W copyright

--copyleft

--copyright

Print the short version of the GNU copyright information message on the standard output and exit successfully.

-W dump-variables[=file]

--dump-variables[=file]

Print a sorted list of global variables, their types and final values to *file*. If no *file* is provided, *gawk* uses a file named **awkvars.out** in the current directory.

Having a list of all the global variables is a good way to look for typographical errors in your programs. You would also use this option if you have a large program with a lot of functions, and you want to be sure that your functions don't inadvertently use global variables that you meant to be local. (This is a particularly easy mistake to make with simple variable names like *i*, *j*, and so on.)

-W exec file

--exec file

Similar to **-f**, however, this option is the last one processed. This should be used with **#!** scripts, particularly for CGI applications, to avoid passing in options or source code (!) on the command line from a URL. This option disables command-line variable assignments.

-W gen-po

--gen-po

Scan and parse the AWK program, and generate a GNU **.po** format file on standard output with entries for all localizable strings in the program. The program itself is not executed. See the GNU *gettext* distribution for more information on **.po** files.

-W help

-W usage

--help

--usage

Print a relatively short summary of the available options on the standard output. (Per the *GNU Coding Standards*, these options cause an immediate, successful exit.)

-W lint[=value]

--lint[=value]

Provide warnings about constructs that are dubious or non-portable to other AWK implementations. With an optional argument of **fatal**, lint warnings become fatal errors. This may be drastic, but its use will certainly encourage the development of cleaner AWK programs. With an optional argument of **invalid**, only warnings about things that are actually invalid are issued. (This is not fully implemented yet.)

-W lint-old

--lint-old

Provide warnings about constructs that are not portable to the original version of Unix *awk*.

-W non-decimal-data

--non-decimal-data

Recognize octal and hexadecimal values in input data. *Use this option with great caution!*

-W posix**--posix**

This turns on *compatibility* mode, with the following additional restrictions:

- `\x` escape sequences are not recognized.
- Only space and tab act as field separators when **FS** is set to a single space, newline does not.
- You cannot continue lines after `?` and `:`.
- The synonym **func** for the keyword **function** is not recognized.
- The operators `**` and `**=` cannot be used in place of `^` and `^=`.
- The **fflush()** function is not available.

-W profile[=prof_file]**--profile[=prof_file]**

Send profiling data to *prof_file*. The default is **awkprof.out**. When run with *gawk*, the profile is just a “pretty printed” version of the program. When run with *pgawk*, the profile contains execution counts of each statement in the program in the left margin and function call counts for each user-defined function.

-W re-interval**--re-interval**

Enable the use of *interval expressions* in regular expression matching (see **Regular Expressions**, below). Interval expressions were not traditionally available in the AWK language. The POSIX standard added them, to make *awk* and *egrep* consistent with each other. However, their use is likely to break old AWK programs, so *gawk* only provides them if they are requested with this option, or when **--posix** is specified.

-W source program-text**--source program-text**

Use *program-text* as AWK program source code. This option allows the easy intermixing of library functions (used via the **-f** and **--file** options) with source code entered on the command line. It is intended primarily for medium to large AWK programs used in shell scripts.

-W use-lc-numeric**--use-lc-numeric**

This forces *gawk* to use the locale’s decimal point character when parsing input data. Although the POSIX standard requires this behavior, and *gawk* does so when **--posix** is in effect, the default is to follow traditional behavior and use a period as the decimal point, even in locales where the period is not the decimal point character. This option overrides the default behavior, without the full draconian strictness of the **--posix** option.

-W version**--version**

Print version information for this particular copy of *gawk* on the standard output. This is useful mainly for knowing if the current copy of *gawk* on your system is up to date with respect to whatever the Free Software Foundation is distributing. This is also useful when reporting bugs. (Per the *GNU Coding Standards*, these options cause an immediate, successful exit.)

--

Signal the end of options. This is useful to allow further arguments to the AWK program itself to start with a “-”. This provides consistency with the argument parsing convention used by most other POSIX programs.

In compatibility mode, any other options are flagged as invalid, but are otherwise ignored. In normal operation, as long as program text has been supplied, unknown options are passed on to the AWK program in the **ARGV** array for processing. This is particularly useful for running AWK programs via the “#!” executable interpreter mechanism.

AWK PROGRAM EXECUTION

An AWK program consists of a sequence of pattern-action statements and optional function definitions.

```
pattern { action statements }
```



```
function name(parameter list) { statements }
```

Gawk first reads the program source from the *program-file*(s) if specified, from arguments to **--source**, or from the first non-option argument on the command line. The **-f** and **--source** options may be used multiple times on the command line. *Gawk* reads the program text as if all the *program-files* and command line source texts had been concatenated together. This is useful for building libraries of AWK functions, without having to include them in each new AWK program that uses them. It also provides the ability to mix library functions with command line programs.

The environment variable **AWKPATH** specifies a search path to use when finding source files named with the **-f** option. If this variable does not exist, the default path is **"/usr/local/share/awk"**. (The actual directory may vary, depending upon how *gawk* was built and installed.) If a file name given to the **-f** option contains a **"/"** character, no path search is performed.

Gawk executes AWK programs in the following order. First, all variable assignments specified via the **-v** option are performed. Next, *gawk* compiles the program into an internal form. Then, *gawk* executes the code in the **BEGIN** block(s) (if any), and then proceeds to read each file named in the **ARGV** array. If there are no files named on the command line, *gawk* reads the standard input.

If a filename on the command line has the form *var=val* it is treated as a variable assignment. The variable *var* will be assigned the value *val*. (This happens after any **BEGIN** block(s) have been run.) Command line variable assignment is most useful for dynamically assigning values to the variables AWK uses to control how input is broken into fields and records. It is also useful for controlling state if multiple passes are needed over a single data file.

If the value of a particular element of **ARGV** is empty (**""**), *gawk* skips over it.

For each record in the input, *gawk* tests to see if it matches any *pattern* in the AWK program. For each pattern that the record matches, the associated *action* is executed. The patterns are tested in the order they occur in the program.

Finally, after all the input is exhausted, *gawk* executes the code in the **END** block(s) (if any).

VARIABLES, RECORDS AND FIELDS

AWK variables are dynamic; they come into existence when they are first used. Their values are either floating-point numbers or strings, or both, depending upon how they are used. AWK also has one dimensional arrays; arrays with multiple dimensions may be simulated. Several pre-defined variables are set as a program runs; these are described as needed and summarized below.

Records

Normally, records are separated by newline characters. You can control how records are separated by assigning values to the built-in variable **RS**. If **RS** is any single character, that character separates records. Otherwise, **RS** is a regular expression. Text in the input that matches this regular expression separates the record. However, in compatibility mode, only the first character of its string value is used for separating records. If **RS** is set to the null string, then records are separated by blank lines. When **RS** is set to the null string, the newline character always acts as a field separator, in addition to whatever value **FS** may have.

Fields

As each input record is read, *gawk* splits the record into *fields*, using the value of the **FS** variable as the field separator. If **FS** is a single character, fields are separated by that character. If **FS** is the null string, then each individual character becomes a separate field. Otherwise, **FS** is expected to be a full regular expression. In the special case that **FS** is a single space, fields are separated by runs of spaces and/or tabs and/or newlines. (But see the section **POSIX COMPATIBILITY**, below). **NOTE:** The value of **IGNORECASE** (see below) also affects how fields are split when **FS** is a regular expression, and how records are separated when **RS** is a regular expression.

If the **FIELDWIDTHS** variable is set to a space separated list of numbers, each field is expected to have fixed width, and *gawk* splits up the record using the specified widths. The value of **FS** is ignored. Assigning a new value to **FS** overrides the use of **FIELDWIDTHS**, and restores the default behavior.

Each field in the input record may be referenced by its position, **\$1**, **\$2**, and so on. **\$0** is the whole record. Fields need not be referenced by constants:

```
n = 5
print $n
```

prints the fifth field in the input record.

The variable **NF** is set to the total number of fields in the input record.

References to non-existent fields (i.e. fields after **\$NF**) produce the null-string. However, assigning to a non-existent field (e.g., **\$(NF+2) = 5**) increases the value of **NF**, creates any intervening fields with the null string as their value, and causes the value of **\$0** to be recomputed, with the fields being separated by the value of **OFS**. References to negative numbered fields cause a fatal error. Decrementing **NF** causes the values of fields past the new value to be lost, and the value of **\$0** to be recomputed, with the fields being separated by the value of **OFS**.

Assigning a value to an existing field causes the whole record to be rebuilt when **\$0** is referenced. Similarly, assigning a value to **\$0** causes the record to be resplit, creating new values for the fields.

Built-in Variables

Gawk's built-in variables are:

| | |
|--------------------|--|
| ARGC | The number of command line arguments (does not include options to <i>gawk</i> , or the program source). |
| ARGIND | The index in ARGV of the current file being processed. |
| ARGV | Array of command line arguments. The array is indexed from 0 to ARGC - 1. Dynamically changing the contents of ARGV can control the files used for data. |
| BINMODE | On non-POSIX systems, specifies use of "binary" mode for all file I/O. Numeric values of 1, 2, or 3, specify that input files, output files, or all files, respectively, should use binary I/O. String values of "r", or "w" specify that input files, or output files, respectively, should use binary I/O. String values of "rw" or "wr" specify that all files should use binary I/O. Any other string value is treated as "rw", but generates a warning message. |
| CONVFMT | The conversion format for numbers, "%.6g", by default. |
| ENVIRON | An array containing the values of the current environment. The array is indexed by the environment variables, each element being the value of that variable (e.g., ENVIRON["HOME"] might be /home/arnold). Changing this array does not affect the environment seen by programs which <i>gawk</i> spawns via redirection or the system() function. |
| ERRNO | If a system error occurs either doing a redirection for getline , during a read for getline , or during a close() , then ERRNO will contain a string describing the error. The value is subject to translation in non-English locales. |
| FIELDWIDTHS | A white-space separated list of fieldwidths. When set, <i>gawk</i> parses the input into fields of fixed width, instead of using the value of the FS variable as the field separator. |
| FILENAME | The name of the current input file. If no files are specified on the command line, the value of FILENAME is "-". However, FILENAME is undefined inside the BEGIN block (unless set by getline). |
| FNR | The input record number in the current input file. |
| FS | The input field separator, a space by default. See Fields , above. |
| IGNORECASE | Controls the case-sensitivity of all regular expression and string operations. If IGNORECASE has a non-zero value, then string comparisons and pattern matching in rules, field splitting with FS , record separating with RS , regular expression matching with ~ and !~ , and the gensub() , gsub() , index() , match() , split() , and sub() built-in functions all ignore case when doing regular expression operations. NOTE : Array subscripting is <i>not</i> affected. However, the asort() and asorti() functions are affected. Thus, if IGNORECASE is not equal to zero, /aB/ matches all of the strings "ab", "aB", "Ab", and "AB". As with all AWK variables, the initial value of IGNORECASE is zero, so all regular expression and string operations are normally case-sensitive. Under Unix, the full ISO 8859-1 Latin-1 character set is used when ignoring case. As of <i>gawk</i> 3.1.4, the case equivalencies are fully locale-aware, based on the C <ctype.h> facilities such as isalpha() , and toupper() . |
| LINT | Provides dynamic control of the --lint option from within an AWK program. When true, <i>gawk</i> prints lint warnings. When false, it does not. When assigned the string value "fatal", lint warnings become fatal errors, exactly like --lint=fatal . Any other true value just prints warnings. |

| | |
|----------------------------|--|
| NF | The number of fields in the current input record. |
| NR | The total number of input records seen so far. |
| OFMT | The output format for numbers, "%.6g" , by default. |
| OFS | The output field separator, a space by default. |
| ORS | The output record separator, by default a newline. |
| PROCINFO | The elements of this array provide access to information about the running AWK program. On some systems, there may be elements in the array, "group1" through "groupn" for some <i>n</i> , which is the number of supplementary groups that the process has. Use the in operator to test for these elements. The following elements are guaranteed to be available: |
| PROCINFO["egid"] | the value of the <i>getegid</i> (2) system call. |
| PROCINFO["euid"] | the value of the <i>geteuid</i> (2) system call. |
| PROCINFO["FS"] | "FS" if field splitting with FS is in effect, or "FIELDWIDTHS" if field splitting with FIELDWIDTHS is in effect. |
| PROCINFO["gid"] | the value of the <i>getgid</i> (2) system call. |
| PROCINFO["pgrp"] | the process group ID of the current process. |
| PROCINFO["pid"] | the process ID of the current process. |
| PROCINFO["ppid"] | the parent process ID of the current process. |
| PROCINFO["uid"] | the value of the <i>getuid</i> (2) system call. |
| PROCINFO["version"] | The version of <i>gawk</i> . This is available from version 3.1.4 and later. |
| RS | The input record separator, by default a newline. |
| RT | The record terminator. <i>Gawk</i> sets RT to the input text that matched the character or regular expression specified by RS . |
| RSTART | The index of the first character matched by match() ; 0 if no match. (This implies that character indices start at one.) |
| RLENGTH | The length of the string matched by match() ; -1 if no match. |
| SUBSEP | The character used to separate multiple subscripts in array elements, by default "\034" . |
| TEXTDOMAIN | The text domain of the AWK program; used to find the localized translations for the program's strings. |

Arrays

Arrays are subscripted with an expression between square brackets ([and]). If the expression is an expression list (*expr*, *expr* ...) then the array subscript is a string consisting of the concatenation of the (string) value of each expression, separated by the value of the **SUBSEP** variable. This facility is used to simulate multiply dimensioned arrays. For example:

```
i = "A"; j = "B"; k = "C"
x[i, j, k] = "hello, world\n"
```

assigns the string **"hello, world\n"** to the element of the array **x** which is indexed by the string **"A\034B\034C"**. All arrays in AWK are associative, i.e. indexed by string values.

The special operator **in** may be used to test if an array has an index consisting of a particular value.

```
if (val in array)
    print array[val]
```

If the array has multiple subscripts, use **(i, j) in array**.

The **in** construct may also be used in a **for** loop to iterate over all the elements of an array.

An element may be deleted from an array using the **delete** statement. The **delete** statement may also be used to delete the entire contents of an array, just by specifying the array name without a subscript.

Variable Typing And Conversion

Variables and fields may be (floating point) numbers, or strings, or both. How the value of a variable is interpreted depends upon its context. If used in a numeric expression, it will be treated as a number; if used as a string it will be treated as a string.

To force a variable to be treated as a number, add 0 to it; to force it to be treated as a string, concatenate it with the null string.

When a string must be converted to a number, the conversion is accomplished using *strtod*(3). A number is converted to a string by using the value of **CONVFMT** as a format string for *sprintf*(3), with the numeric value of the variable as the argument. However, even though all numbers in AWK are floating-point, integral values are *always* converted as integers. Thus, given

```
CONVFMT = "%2.2f"
```

```
a = 12
```

```
b = a ""
```

the variable **b** has a string value of **"12"** and not **"12.00"**.

When operating in POSIX mode (such as with the **--posix** command line option), beware that locale settings may interfere with the way decimal numbers are treated: the decimal separator of the numbers you are feeding to *gawk* must conform to what your locale would expect, be it a comma (,) or a period (.).

Gawk performs comparisons as follows: If two variables are numeric, they are compared numerically. If one value is numeric and the other has a string value that is a “numeric string,” then comparisons are also done numerically. Otherwise, the numeric value is converted to a string and a string comparison is performed. Two strings are compared, of course, as strings.

Note that string constants, such as **"57"**, are *not* numeric strings, they are string constants. The idea of “numeric string” only applies to fields, **getline** input, **FILENAME**, **ARGV** elements, **ENVIRON** elements and the elements of an array created by **split()** that are numeric strings. The basic idea is that *user input*, and only user input, that looks numeric, should be treated that way.

Uninitialized variables have the numeric value 0 and the string value "" (the null, or empty, string).

Octal and Hexadecimal Constants

Starting with version 3.1 of *gawk*, you may use C-style octal and hexadecimal constants in your AWK program source code. For example, the octal value **011** is equal to decimal **9**, and the hexadecimal value **0x11** is equal to decimal 17.

String Constants

String constants in AWK are sequences of characters enclosed between double quotes ("). Within strings, certain *escape sequences* are recognized, as in C. These are:

**** A literal backslash.

\a The “alert” character; usually the ASCII BEL character.

\b backspace.

\f form-feed.

\n newline.

\r carriage return.

\t horizontal tab.

\v vertical tab.

\xhex digits

The character represented by the string of hexadecimal digits following the **\x**. As in ANSI C, all following hexadecimal digits are considered part of the escape sequence. (This feature should tell us something about language design by committee.) E.g., **"\x1B"** is the ASCII ESC (escape) character.

\ddd The character represented by the 1-, 2-, or 3-digit sequence of octal digits. E.g., **"\033"** is the ASCII ESC (escape) character.

\c The literal character *c*.

The escape sequences may also be used inside constant regular expressions (e.g., **/[\t\n\r\v]/** matches whitespace characters).

In compatibility mode, the characters represented by octal and hexadecimal escape sequences are treated literally when used in regular expression constants. Thus, **/a\52b/** is equivalent to **/a*b/**.

PATTERNS AND ACTIONS

AWK is a line-oriented language. The pattern comes first, and then the action. Action statements are enclosed in { and }. Either the pattern may be missing, or the action may be missing, but, of course, not both. If the pattern is missing, the action is executed for every single record of input. A missing action is equivalent to

```
{ print }
```

which prints the entire record.

Comments begin with the “#” character, and continue until the end of the line. Blank lines may be used to

separate statements. Normally, a statement ends with a newline, however, this is not the case for lines ending in a “;”, {, ?, :, &&, or ||. Lines ending in **do** or **else** also have their statements automatically continued on the following line. In other cases, a line can be continued by ending it with a “\”, in which case the new-line will be ignored.

Multiple statements may be put on one line by separating them with a “;”. This applies to both the statements within the action part of a pattern-action pair (the usual case), and to the pattern-action statements themselves.

Patterns

AWK patterns may be one of the following:

```
BEGIN
END
/regular expression/
relational expression
pattern && pattern
pattern || pattern
pattern ? pattern : pattern
(pattern)
!pattern
pattern1, pattern2
```

BEGIN and **END** are two special kinds of patterns which are not tested against the input. The action parts of all **BEGIN** patterns are merged as if all the statements had been written in a single **BEGIN** block. They are executed before any of the input is read. Similarly, all the **END** blocks are merged, and executed when all the input is exhausted (or when an **exit** statement is executed). **BEGIN** and **END** patterns cannot be combined with other patterns in pattern expressions. **BEGIN** and **END** patterns cannot have missing action parts.

For */regular expression/* patterns, the associated statement is executed for each input record that matches the regular expression. Regular expressions are the same as those in *egrep*(1), and are summarized below.

A *relational expression* may use any of the operators defined below in the section on actions. These generally test whether certain fields match certain regular expressions.

The **&&**, **||**, and **!** operators are logical AND, logical OR, and logical NOT, respectively, as in C. They do short-circuit evaluation, also as in C, and are used for combining more primitive pattern expressions. As in most languages, parentheses may be used to change the order of evaluation.

The **?:** operator is like the same operator in C. If the first pattern is true then the pattern used for testing is the second pattern, otherwise it is the third. Only one of the second and third patterns is evaluated.

The *pattern1, pattern2* form of an expression is called a *range pattern*. It matches all input records starting with a record that matches *pattern1*, and continuing until a record that matches *pattern2*, inclusive. It does not combine with any other sort of pattern expression.

Regular Expressions

Regular expressions are the extended kind found in *egrep*. They are composed of characters as follows:

| | |
|------------------|--|
| <i>c</i> | matches the non-metacharacter <i>c</i> . |
| <i>\c</i> | matches the literal character <i>c</i> . |
| <i>.</i> | matches any character <i>including</i> newline. |
| <i>^</i> | matches the beginning of a string. |
| <i>\$</i> | matches the end of a string. |
| <i>[abc...]</i> | character list, matches any of the characters <i>abc...</i> |
| <i>[^abc...]</i> | negated character list, matches any character except <i>abc...</i> |
| <i>r1 r2</i> | alternation: matches either <i>r1</i> or <i>r2</i> . |
| <i>r1r2</i> | concatenation: matches <i>r1</i> , and then <i>r2</i> . |
| <i>r+</i> | matches one or more <i>r</i> 's. |
| <i>r*</i> | matches zero or more <i>r</i> 's. |
| <i>r?</i> | matches zero or one <i>r</i> 's. |
| <i>(r)</i> | grouping: matches <i>r</i> . |
| <i>r{n}</i> | |

| | |
|--------------------|---|
| $r\{n,\}$ | |
| $r\{n,m\}$ | One or two numbers inside braces denote an <i>interval expression</i> . If there is one number in the braces, the preceding regular expression r is repeated n times. If there are two numbers separated by a comma, r is repeated n to m times. If there is one number followed by a comma, then r is repeated at least n times. |
| | Interval expressions are only available if either --posix or --re-interval is specified on the command line. |
| <code>\y</code> | matches the empty string at either the beginning or the end of a word. |
| <code>\B</code> | matches the empty string within a word. |
| <code>\<</code> | matches the empty string at the beginning of a word. |
| <code>\></code> | matches the empty string at the end of a word. |
| <code>\w</code> | matches any word-constituent character (letter, digit, or underscore). |
| <code>\W</code> | matches any character that is not word-constituent. |
| <code>\‘</code> | matches the empty string at the beginning of a buffer (string). |
| <code>\’</code> | matches the empty string at the end of a buffer. |

The escape sequences that are valid in string constants (see below) are also valid in regular expressions.

Character classes are a feature introduced in the POSIX standard. A character class is a special notation for describing lists of characters that have a specific attribute, but where the actual characters themselves can vary from country to country and/or from character set to character set. For example, the notion of what is an alphabetic character differs in the USA and in France.

A character class is only valid in a regular expression *inside* the brackets of a character list. Character classes consist of `[:`, a keyword denoting the class, and `:]`. The character classes defined by the POSIX standard are:

| | |
|--------------------------------|---|
| <code>[:alnum:]</code> | Alphanumeric characters. |
| <code>[:alpha:]</code> | Alphabetic characters. |
| <code>[:blank:]</code> | Space or tab characters. |
| <code>[:cntrl:]</code> | Control characters. |
| <code>[:digit:]</code> | Numeric characters. |
| <code>[:graph:]</code> | Characters that are both printable and visible. (A space is printable, but not visible, while an a is both.) |
| <code>[:lower:]</code> | Lower-case alphabetic characters. |
| <code>[:print:]</code> | Printable characters (characters that are not control characters.) |
| <code>[:punct:]</code> | Punctuation characters (characters that are not letter, digits, control characters, or space characters). |
| <code>[:space:]</code> | Space characters (such as space, tab, and formfeed, to name a few). |
| <code>[:upper:]</code> | Upper-case alphabetic characters. |
| <code>[:xdigit:]</code> | Characters that are hexadecimal digits. |

For example, before the POSIX standard, to match alphanumeric characters, you would have had to write `/[A-Za-z0-9]/`. If your character set had other alphabetic characters in it, this would not match them, and if your character set collated differently from ASCII, this might not even match the ASCII alphanumeric characters. With the POSIX character classes, you can write `/[[:alnum:]]/`, and this matches the alphabetic and numeric characters in your character set, no matter what it is.

Two additional special sequences can appear in character lists. These apply to non-ASCII character sets, which can have single symbols (called *collating elements*) that are represented with more than one

character, as well as several characters that are equivalent for *collating*, or sorting, purposes. (E.g., in French, a plain “e” and a grave-accented “è” are equivalent.)

Collating Symbols

A collating symbol is a multi-character collating element enclosed in [. and .]. For example, if **ch** is a collating element, then **[[.ch.]]** is a regular expression that matches this collating element, while **[ch]** is a regular expression that matches either **c** or **h**.

Equivalence Classes

An equivalence class is a locale-specific name for a list of characters that are equivalent. The name is enclosed in [= and =]. For example, the name **e** might be used to represent all of “e,” “é,” and “è.” In this case, **[[=e=]]** is a regular expression that matches any of **e**, **é**, or **è**.

These features are very valuable in non-English speaking locales. The library functions that *gawk* uses for regular expression matching currently only recognize POSIX character classes; they do not recognize collating symbols or equivalence classes.

The **\y**, **\B**, **\<**, **\>**, **\w**, **\W**, **\'**, and **\'** operators are specific to *gawk*; they are extensions based on facilities in the GNU regular expression libraries.

The various command line options control how *gawk* interprets characters in regular expressions.

No options

In the default case, *gawk* provide all the facilities of POSIX regular expressions and the GNU regular expression operators described above. However, interval expressions are not supported.

--posix

Only POSIX regular expressions are supported, the GNU operators are not special. (E.g., **\w** matches a literal **w**). Interval expressions are allowed.

--traditional

Traditional Unix *awk* regular expressions are matched. The GNU operators are not special, interval expressions are not available, and neither are the POSIX character classes (**[[:alnum:]]** and so on). Characters described by octal and hexadecimal escape sequences are treated literally, even if they represent regular expression metacharacters.

--re-interval

Allow interval expressions in regular expressions, even if **--traditional** has been provided.

Actions

Action statements are enclosed in braces, { and }. Action statements consist of the usual assignment, conditional, and looping statements found in most languages. The operators, control statements, and input/output statements available are patterned after those in C.

Operators

The operators in AWK, in order of decreasing precedence, are

| | |
|-------|--|
| (...) | Grouping |
| \$ | Field reference. |
| ++ -- | Increment and decrement, both prefix and postfix. |
| ^ | Exponentiation (** may also be used, and **= for the assignment operator). |
| + - ! | Unary plus, unary minus, and logical negation. |
| * / % | Multiplication, division, and modulus. |
| + - | Addition and subtraction. |
| space | String concatenation. |
| & | Piped I/O for getline , print , and printf . |
| < > | |

| | |
|--------------------------|---|
| <code><= >=</code> | |
| <code>!= ==</code> | The regular relational operators. |
| <code>~ !~</code> | Regular expression match, negated match. NOTE: Do not use a constant regular expression (<code>/foo/</code>) on the left-hand side of a <code>~</code> or <code>!~</code> . Only use one on the right-hand side. The expression <code>/foo/ ~ exp</code> has the same meaning as <code>((<code>\$0</code> ~ /foo/) ~ exp)</code> . This is usually <i>not</i> what was intended. |
| <code>in</code> | Array membership. |
| <code>&&</code> | Logical AND. |
| <code> </code> | Logical OR. |
| <code>?:</code> | The C conditional expression. This has the form <code>expr1 ? expr2 : expr3</code> . If <code>expr1</code> is true, the value of the expression is <code>expr2</code> , otherwise it is <code>expr3</code> . Only one of <code>expr2</code> and <code>expr3</code> is evaluated. |
| <code>= += -=</code> | |
| <code>*= /= %= ^=</code> | Assignment. Both absolute assignment (<code>var = value</code>) and operator-assignment (the other forms) are supported. |

Control Statements

The control statements are as follows:

```

if (condition) statement [ else statement ]
while (condition) statement
do statement while (condition)
for (expr1; expr2; expr3) statement
for (var in array) statement
break
continue
delete array[index]
delete array
exit [ expression ]
{ statements }

```

I/O Statements

The input/output statements are as follows:

| | |
|---|---|
| close (<i>file</i> [, <i>how</i>]) | Close file, pipe or co-process. The optional <i>how</i> should only be used when closing one end of a two-way pipe to a co-process. It must be a string value, either "to" or "from" . |
| getline | Set \$0 from next input record; set NF , NR , FNR . |
| getline < <i>file</i> | Set \$0 from next record of <i>file</i> ; set NF . |
| getline <i>var</i> | Set <i>var</i> from next input record; set NR , FNR . |
| getline <i>var</i> < <i>file</i> | Set <i>var</i> from next record of <i>file</i> . |
| command getline [<i>var</i>] | Run <i>command</i> piping the output either into \$0 or <i>var</i> , as above. |
| command & getline [<i>var</i>] | Run <i>command</i> as a co-process piping the output either into \$0 or <i>var</i> , as above. Co-processes are a <i>gawk</i> extension. (<i>command</i> can also be a socket. See the subsection Special File Names , below.) |
| next | Stop processing the current input record. The next input record is read and processing starts over with the first pattern in the AWK program. If the end of the input data is reached, the END block(s), if any, are executed. |

| | |
|---|---|
| nextfile | Stop processing the current input file. The next input record read comes from the next input file. FILENAME and ARGIND are updated, FNR is reset to 1, and processing starts over with the first pattern in the AWK program. If the end of the input data is reached, the END block(s), if any, are executed. |
| print | Prints the current record. The output record is terminated with the value of the ORS variable. |
| print <i>expr-list</i> | Prints expressions. Each expression is separated by the value of the OFS variable. The output record is terminated with the value of the ORS variable. |
| print <i>expr-list</i> > <i>file</i> | Prints expressions on <i>file</i> . Each expression is separated by the value of the OFS variable. The output record is terminated with the value of the ORS variable. |
| printf <i>fmt</i> , <i>expr-list</i> | Format and print. |
| printf <i>fmt</i> , <i>expr-list</i> > <i>file</i> | Format and print on <i>file</i> . |
| system (<i>cmd-line</i>) | Execute the command <i>cmd-line</i> , and return the exit status. (This may not be available on non-POSIX systems.) |
| fflush ([<i>file</i>]) | Flush any buffers associated with the open output file or pipe <i>file</i> . If <i>file</i> is missing, then standard output is flushed. If <i>file</i> is the null string, then all open output files and pipes have their buffers flushed. |

Additional output redirections are allowed for **print** and **printf**.

print ... >> *file*
Appends output to the *file*.

print ... | *command*
Writes on a pipe.

print ... |& *command*
Sends data to a co-process or socket. (See also the subsection **Special File Names**, below.)

The **getline** command returns 0 on end of file and -1 on an error. Upon an error, **ERRNO** contains a string describing the problem.

NOTE: If using a pipe, co-process, or socket to **getline**, or from **print** or **printf** within a loop, you *must* use **close()** to create new instances of the command or socket. AWK does not automatically close pipes, sockets, or co-processes when they return EOF.

The **printf** Statement

The AWK versions of the **printf** statement and **sprintf()** function (see below) accept the following conversion specification formats:

| | |
|-----------------------|--|
| %c | An ASCII character. If the argument used for %c is numeric, it is treated as a character and printed. Otherwise, the argument is assumed to be a string, and the only first character of that string is printed. |
| %d , %i | A decimal number (the integer part). |
| %e , %E | A floating point number of the form [-]d.dddde[+-]dd . The %E format uses E instead of e . |
| %f , %F | A floating point number of the form [-]ddd.ddd . If the system library supports it, %F is available as well. This is like %f , but uses capital letters for special “not a number” and “infinity” values. If %F is not available, <i>gawk</i> uses %f . |
| %g , %G | Use %e or %f conversion, whichever is shorter, with nonsignificant zeros suppressed. The %G format uses %E instead of %e . |

- %o** An unsigned octal number (also an integer).
- %u** An unsigned decimal number (again, an integer).
- %s** A character string.
- %x, %X** An unsigned hexadecimal number (an integer). The **%X** format uses **ABCDEF** instead of **abcdef**.
- %%** A single **%** character; no argument is converted.

NOTE: When using the integer format-control letters for values that are outside the range of a C **long** integer, *gawk* switches to the **%0f** format specifier. If **--lint** is provided on the command line *gawk* warns about this. Other versions of *awk* may print invalid values or do something else entirely.

Optional, additional parameters may lie between the **%** and the control letter:

- count\$** Use the *count*'th argument at this point in the formatting. This is called a *positional specifier* and is intended primarily for use in translated versions of format strings, not in the original text of an AWK program. It is a *gawk* extension.
- The expression should be left-justified within its field.
- space** For numeric conversions, prefix positive values with a space, and negative values with a minus sign.
- +** The plus sign, used before the width modifier (see below), says to always supply a sign for numeric conversions, even if the data to be formatted is positive. The **+** overrides the space modifier.
- #** Use an "alternate form" for certain control letters. For **%o**, supply a leading zero. For **%x**, and **%X**, supply a leading **0x** or **0X** for a nonzero result. For **%e**, **%E**, **%f** and **%F**, the result always contains a decimal point. For **%g**, and **%G**, trailing zeros are not removed from the result.
- 0** A leading **0** (zero) acts as a flag, that indicates output should be padded with zeroes instead of spaces. This applies even to non-numeric output formats. This flag only has an effect when the field width is wider than the value to be printed.
- width** The field should be padded to this width. The field is normally padded with spaces. If the **0** flag has been used, it is padded with zeroes.
- .prec** A number that specifies the precision to use when printing. For the **%e**, **%E**, **%f** and **%F**, formats, this specifies the number of digits you want printed to the right of the decimal point. For the **%g**, and **%G** formats, it specifies the maximum number of significant digits. For the **%d**, **%o**, **%i**, **%u**, **%x**, and **%X** formats, it specifies the minimum number of digits to print. For **%s**, it specifies the maximum number of characters from the string that should be printed.

The dynamic *width* and *prec* capabilities of the ANSI C **printf()** routines are supported. A ***** in place of either the **width** or **prec** specifications causes their values to be taken from the argument list to **printf** or **sprintf()**. To use a positional specifier with a dynamic width or precision, supply the *count\$* after the ***** in the format string. For example, **"%3\$*2\$.*1\$s"**.

Special File Names

When doing I/O redirection from either **print** or **printf** into a file, or via **getline** from a file, *gawk* recognizes certain special filenames internally. These filenames allow access to open file descriptors inherited from *gawk*'s parent process (usually the shell). These file names may also be used on the command line to name data files. The filenames are:

- /dev/stdin** The standard input.
- /dev/stdout** The standard output.
- /dev/stderr** The standard error output.
- /dev/fd/*n*** The file associated with the open file descriptor *n*.

These are particularly useful for error messages. For example:

```
print "You blew it!" > "/dev/stderr"
```

whereas you would otherwise have to use

```
print "You blew it!" | "cat 1>&2"
```

The following special filenames may be used with the `|&` co-process operator for creating TCP/IP network connections.

/inet/tcp/lport/rhost/rport File for TCP/IP connection on local port *lport* to remote host *rhost* on remote port *rport*. Use a port of **0** to have the system pick a port.

/inet/udp/lport/rhost/rport Similar, but use UDP/IP instead of TCP/IP.

/inet/raw/lport/rhost/rport Reserved for future use.

Other special filenames provide access to information about the running *gawk* process. **These filenames are now obsolete.** Use the **PROCINFO** array to obtain the information they provide. The filenames are:

/dev/pid Reading this file returns the process ID of the current process, in decimal, terminated with a newline.

/dev/ppid Reading this file returns the parent process ID of the current process, in decimal, terminated with a newline.

/dev/pgrp Reading this file returns the process group ID of the current process, in decimal, terminated with a newline.

/dev/user Reading this file returns a single record terminated with a newline. The fields are separated with spaces. **\$1** is the value of the *getuid(2)* system call, **\$2** is the value of the *geteuid(2)* system call, **\$3** is the value of the *getgid(2)* system call, and **\$4** is the value of the *getegid(2)* system call. If there are any additional fields, they are the group IDs returned by *getgroups(2)*. Multiple groups may not be supported on all systems.

Numeric Functions

AWK has the following built-in arithmetic functions:

atan2(y, x) Returns the arctangent of *y/x* in radians.

cos(expr) Returns the cosine of *expr*, which is in radians.

exp(expr) The exponential function.

int(expr) Truncates to integer.

log(expr) The natural logarithm function.

rand() Returns a random number *N*, between 0 and 1, such that $0 \leq N < 1$.

sin(expr) Returns the sine of *expr*, which is in radians.

sqr(expr) The square root function.

srand([expr]) Uses *expr* as a new seed for the random number generator. If no *expr* is provided, the time of day is used. The return value is the previous seed for the random number generator.

String Functions

Gawk has the following built-in string functions:

asort(s [, d]) Returns the number of elements in the source array *s*. The contents of *s* are sorted using *gawk*'s normal rules for comparing values, and the indices of the sorted values of *s* are replaced with sequential integers starting with 1. If the optional destination array *d* is specified, then *s* is first duplicated into *d*, and then *d* is sorted, leaving the indices of the source array *s* unchanged.

asorti(s [, d]) Returns the number of elements in the source array *s*. The behavior is the same as that of **asort()**, except that the array *indices* are used for sorting, not the array values. When done, the array is indexed numerically, and the values are those of the

original indices. The original values are lost; thus provide a second array if you wish to preserve the original.

| | |
|---|--|
| gensub (<i>r</i> , <i>s</i> , <i>h</i> [, <i>t</i>]) | Search the target string <i>t</i> for matches of the regular expression <i>r</i> . If <i>h</i> is a string beginning with g or G , then replace all matches of <i>r</i> with <i>s</i> . Otherwise, <i>h</i> is a number indicating which match of <i>r</i> to replace. If <i>t</i> is not supplied, \$0 is used instead. Within the replacement text <i>s</i> , the sequence \n , where <i>n</i> is a digit from 1 to 9, may be used to indicate just the text that matched the <i>n</i> 'th parenthesized subexpression. The sequence \0 represents the entire matched text, as does the character & . Unlike sub() and gsub() , the modified string is returned as the result of the function, and the original target string is <i>not</i> changed. |
| gsub (<i>r</i> , <i>s</i> [, <i>t</i>]) | For each substring matching the regular expression <i>r</i> in the string <i>t</i> , substitute the string <i>s</i> , and return the number of substitutions. If <i>t</i> is not supplied, use \$0 . An & in the replacement text is replaced with the text that was actually matched. Use \& to get a literal & . (This must be typed as "\&" ; see <i>GAWK: Effective AWK Programming</i> for a fuller discussion of the rules for & 's and backslashes in the replacement text of sub() , gsub() , and gensub() .) |
| index (<i>s</i> , <i>t</i>) | Returns the index of the string <i>t</i> in the string <i>s</i> , or 0 if <i>t</i> is not present. (This implies that character indices start at one.) |
| length ([<i>s</i>]) | Returns the length of the string <i>s</i> , or the length of \$0 if <i>s</i> is not supplied. Starting with version 3.1.5, as a non-standard extension, with an array argument, length() returns the number of elements in the array. |
| match (<i>s</i> , <i>r</i> [, <i>a</i>]) | Returns the position in <i>s</i> where the regular expression <i>r</i> occurs, or 0 if <i>r</i> is not present, and sets the values of RSTART and RLENGTH . Note that the argument order is the same as for the ~ operator: <i>str ~ re</i> . If array <i>a</i> is provided, <i>a</i> is cleared and then elements 1 through <i>n</i> are filled with the portions of <i>s</i> that match the corresponding parenthesized subexpression in <i>r</i> . The 0'th element of <i>a</i> contains the portion of <i>s</i> matched by the entire regular expression <i>r</i> . Subscripts a[n, "start"] , and a[n, "length"] provide the starting index in the string and length respectively, of each matching substring. |
| split (<i>s</i> , <i>a</i> [, <i>r</i>]) | Splits the string <i>s</i> into the array <i>a</i> on the regular expression <i>r</i> , and returns the number of fields. If <i>r</i> is omitted, FS is used instead. The array <i>a</i> is cleared first. Splitting behaves identically to field splitting, described above. |
| sprintf (<i>fnt</i> , <i>expr-list</i>) | Prints <i>expr-list</i> according to <i>fnt</i> , and returns the resulting string. |
| strtonum (<i>str</i>) | Examines <i>str</i> , and returns its numeric value. If <i>str</i> begins with a leading 0 , strtonum() assumes that <i>str</i> is an octal number. If <i>str</i> begins with a leading 0x or 0X , strtonum() assumes that <i>str</i> is a hexadecimal number. |
| sub (<i>r</i> , <i>s</i> [, <i>t</i>]) | Just like gsub() , but only the first matching substring is replaced. |
| substr (<i>s</i> , <i>i</i> [, <i>n</i>]) | Returns the at most <i>n</i> -character substring of <i>s</i> starting at <i>i</i> . If <i>n</i> is omitted, the rest of <i>s</i> is used. |
| tolower (<i>str</i>) | Returns a copy of the string <i>str</i> , with all the upper-case characters in <i>str</i> translated to their corresponding lower-case counterparts. Non-alphabetic characters are left unchanged. |
| toupper (<i>str</i>) | Returns a copy of the string <i>str</i> , with all the lower-case characters in <i>str</i> translated to their corresponding upper-case counterparts. Non-alphabetic characters are left unchanged. |

As of version 3.1.5, *gawk* is multibyte aware. This means that **index()**, **length()**, **substr()** and **match()** all work in terms of characters, not bytes.

Time Functions

Since one of the primary uses of AWK programs is processing log files that contain time stamp information, *gawk* provides the following functions for obtaining time stamps and formatting them.

mktime(*datespec*)

Turns *datespec* into a time stamp of the same form as returned by **sysptime()**. The *datespec* is a string of the form *YYYY MM DD HH MM SS[DST]*. The contents of the string are six or seven numbers representing respectively the full year including century, the month from 1 to 12, the day of the month from 1 to 31, the hour of the day from 0 to 23, the minute from 0 to 59, and the second from 0 to 60, and an optional daylight saving flag. The values of these numbers need not be within the ranges specified; for example, an hour of -1 means 1 hour before midnight. The origin-zero Gregorian calendar is assumed, with year 0 preceding year 1 and year -1 preceding year 0. The time is assumed to be in the local timezone. If the daylight saving flag is positive, the time is assumed to be daylight saving time; if zero, the time is assumed to be standard time; and if negative (the default), **mktime()** attempts to determine whether daylight saving time is in effect for the specified time. If *datespec* does not contain enough elements or if the resulting time is out of range, **mktime()** returns -1.

strftime([*format* [, *timestamp* [, *utc-flag*]])

Formats *timestamp* according to the specification in *format*. If *utc-flag* is present and is non-zero or non-null, the result is in UTC, otherwise the result is in local time. The *timestamp* should be of the same form as returned by **sysptime()**. If *timestamp* is missing, the current time of day is used. If *format* is missing, a default format equivalent to the output of *date*(1) is used. See the specification for the **strftime()** function in ANSI C for the format conversions that are guaranteed to be available.

sysptime() Returns the current time of day as the number of seconds since the Epoch (1970-01-01 00:00:00 UTC on POSIX systems).

Bit Manipulations Functions

Starting with version 3.1 of *gawk*, the following bit manipulation functions are available. They work by converting double-precision floating point values to **uintmax_t** integers, doing the operation, and then converting the result back to floating point. The functions are:

- and(*v1*, *v2*)** Return the bitwise AND of the values provided by *v1* and *v2*.
- compl(*val*)** Return the bitwise complement of *val*.
- lshift(*val*, *count*)** Return the value of *val*, shifted left by *count* bits.
- or(*v1*, *v2*)** Return the bitwise OR of the values provided by *v1* and *v2*.
- rshift(*val*, *count*)** Return the value of *val*, shifted right by *count* bits.
- xor(*v1*, *v2*)** Return the bitwise XOR of the values provided by *v1* and *v2*.

Internationalization Functions

Starting with version 3.1 of *gawk*, the following functions may be used from within your AWK program for translating strings at run-time. For full details, see *GAWK: Effective AWK Programming*.

bindtextdomain(*directory* [, *domain*])

Specifies the directory where *gawk* looks for the **.mo** files, in case they will not or cannot be placed in the “standard” locations (e.g., during testing). It returns the directory where *domain* is “bound.”

The default *domain* is the value of **TEXTDOMAIN**. If *directory* is the null string (“”), then **bindtextdomain()** returns the current binding for the given *domain*.

dcgettext(*string* [, *domain* [, *category*]])

Returns the translation of *string* in text domain *domain* for locale category *category*. The default value for *domain* is the current value of **TEXTDOMAIN**. The default value for *category* is **“LC_MESSAGES”**.

If you supply a value for *category*, it must be a string equal to one of the known locale categories described in *GAWK: Effective AWK Programming*. You must also supply a text domain. Use **TEXTDOMAIN** if you want to use the current domain.

dcgettext(*string1* , *string2* , *number* [, *domain* [, *category*]])

Returns the plural form used for *number* of the translation of *string1* and *string2* in text domain *domain* for locale category *category*. The default value for *domain* is the current value of **TEXTDOMAIN**. The default value for *category* is "LC_MESSAGES".

If you supply a value for *category*, it must be a string equal to one of the known locale categories described in *GAWK: Effective AWK Programming*. You must also supply a text domain. Use **TEXTDOMAIN** if you want to use the current domain.

USER-DEFINED FUNCTIONS

Functions in AWK are defined as follows:

```
function name(parameter list) { statements }
```

Functions are executed when they are called from within expressions in either patterns or actions. Actual parameters supplied in the function call are used to instantiate the formal parameters declared in the function. Arrays are passed by reference, other variables are passed by value.

Since functions were not originally part of the AWK language, the provision for local variables is rather clumsy: They are declared as extra parameters in the parameter list. The convention is to separate local variables from real parameters by extra spaces in the parameter list. For example:

```
function f(p, q,  a, b)  # a and b are local
{
    ...
}

/abc/  { ... ; f(1, 2) ; ... }
```

The left parenthesis in a function call is required to immediately follow the function name, without any intervening white space. This avoids a syntactic ambiguity with the concatenation operator. This restriction does not apply to the built-in functions listed above.

Functions may call each other and may be recursive. Function parameters used as local variables are initialized to the null string and the number zero upon function invocation.

Use **return** *expr* to return a value from a function. The return value is undefined if no value is provided, or if the function returns by “falling off” the end.

If **--lint** has been provided, *gawk* warns about calls to undefined functions at parse time, instead of at run time. Calling an undefined function at run time is a fatal error.

The word **func** may be used in place of **function**.

DYNAMICALLY LOADING NEW FUNCTIONS

Beginning with version 3.1 of *gawk*, you can dynamically add new built-in functions to the running *gawk* interpreter. The full details are beyond the scope of this manual page; see *GAWK: Effective AWK Programming* for the details.

extension(*object*, *function*)

Dynamically link the shared object file named by *object*, and invoke *function* in that object, to perform initialization. These should both be provided as strings. Returns the value returned by *function*.

This function is provided and documented in *GAWK: Effective AWK Programming*, but everything about this feature is likely to change eventually. We STRONGLY recommend that you do not use this feature for anything that you aren’t willing to redo.

SIGNALS

pgawk accepts two signals. **SIGUSR1** causes it to dump a profile and function call stack to the profile file, which is either **awkprof.out**, or whatever file was named with the **--profile** option. It then continues to run. **SIGHUP** causes *pgawk* to dump the profile and function call stack and then exit.

EXAMPLES

Print and sort the login names of all users:

```
BEGIN { FS = ":" }
      { print $1 | "sort" }
```

Count lines in a file:

```
      { nlines++ }
END   { print nlines }
```

Precede each line by its number in the file:

```
      { print FNR, $0 }
```

Concatenate and line number (a variation on a theme):

```
      { print NR, $0 }
```

Run an external command for particular lines of data:

```
tail -f access_log |
awk 'myhome.html/ { system("nmap " $1 ">> logdir/myhome.html") }'
```

INTERNATIONALIZATION

String constants are sequences of characters enclosed in double quotes. In non-English speaking environments, it is possible to mark strings in the AWK program as requiring translation to the native natural language. Such strings are marked in the AWK program with a leading underscore (“_”). For example,

```
gawk 'BEGIN { print "hello, world" }'
```

always prints **hello, world**. But,

```
gawk 'BEGIN { print _"hello, world" }'
```

might print **bonjour, monde** in France.

There are several steps involved in producing and running a localizable AWK program.

1. Add a **BEGIN** action to assign a value to the **TEXTDOMAIN** variable to set the text domain to a name associated with your program.

```
BEGIN { TEXTDOMAIN = "myprog" }
```

This allows *gawk* to find the **.mo** file associated with your program. Without this step, *gawk* uses the **messages** text domain, which likely does not contain translations for your program.

2. Mark all strings that should be translated with leading underscores.
3. If necessary, use the **dcgettext()** and/or **bindtextdomain()** functions in your program, as appropriate.
4. Run **gawk --gen-po -f myprog.awk > myprog.po** to generate a **.po** file for your program.
5. Provide appropriate translations, and build and install the corresponding **.mo** files.

The internationalization features are described in full detail in *GAWK: Effective AWK Programming*.

POSIX COMPATIBILITY

A primary goal for *gawk* is compatibility with the POSIX standard, as well as with the latest version of UNIX *awk*. To this end, *gawk* incorporates the following user visible features which are not described in the AWK book, but are part of the Bell Laboratories version of *awk*, and are in the POSIX standard.

The book indicates that command line variable assignment happens when *awk* would otherwise open the argument as a file, which is after the **BEGIN** block is executed. However, in earlier implementations, when such an assignment appeared before any file names, the assignment would happen *before* the **BEGIN** block was run. Applications came to depend on this “feature.” When *awk* was changed to match its documentation, the **-v** option for assigning variables before program execution was added to accommodate applications that depended upon the old behavior. (This feature was agreed upon by both the Bell Laboratories and the GNU developers.)

The **-W** option for implementation specific features is from the POSIX standard.

When processing arguments, *gawk* uses the special option “--” to signal the end of arguments. In compatibility mode, it warns about but otherwise ignores undefined options. In normal operation, such arguments are passed on to the AWK program for it to process.

The AWK book does not define the return value of **srand()**. The POSIX standard has it return the seed it was using, to allow keeping track of random number sequences. Therefore **srand()** in *gawk* also returns its current seed.

Other new features are: The use of multiple **-f** options (from MKS *awk*); the **ENVIRON** array; the **\a**, and **\v** escape sequences (done originally in *gawk* and fed back into the Bell Laboratories version); the **tolower()** and **toupper()** built-in functions (from the Bell Laboratories version); and the ANSI C conversion specifications in **printf** (done first in the Bell Laboratories version).

HISTORICAL FEATURES

There are two features of historical AWK implementations that *gawk* supports. First, it is possible to call the **length()** built-in function not only with no argument, but even without parentheses! Thus,

```
a = length      # Holy Algol 60, Batman!
```

is the same as either of

```
a = length()
a = length($0)
```

This feature is marked as “deprecated” in the POSIX standard, and *gawk* issues a warning about its use if **--lint** is specified on the command line.

The other feature is the use of either the **continue** or the **break** statements outside the body of a **while**, **for**, or **do** loop. Traditional AWK implementations have treated such usage as equivalent to the **next** statement. *Gawk* supports this usage if **--traditional** has been specified.

GNU EXTENSIONS

Gawk has a number of extensions to POSIX *awk*. They are described in this section. All the extensions described here can be disabled by invoking *gawk* with the **--traditional** or **--posix** options.

The following features of *gawk* are not available in POSIX *awk*.

- No path search is performed for files named via the **-f** option. Therefore the **AWKPATH** environment variable is not special.
- The **\x** escape sequence. (Disabled with **--posix**.)
- The **fflush()** function. (Disabled with **--posix**.)
- The ability to continue lines after **?** and **:**. (Disabled with **--posix**.)
- Octal and hexadecimal constants in AWK programs.
- The **ARGIND**, **BINMODE**, **ERRNO**, **LINT**, **RT** and **TEXTDOMAIN** variables are not special.

- The **IGNORECASE** variable and its side-effects are not available.
- The **FIELDWIDTHS** variable and fixed-width field splitting.
- The **PROCINFO** array is not available.
- The use of **RS** as a regular expression.
- The special file names available for I/O redirection are not recognized.
- The **|&** operator for creating co-processes.
- The ability to split out individual characters using the null string as the value of **FS**, and as the third argument to **split()**.
- The optional second argument to the **close()** function.
- The optional third argument to the **match()** function.
- The ability to use positional specifiers with **printf** and **sprintf()**.
- The use of **delete array** to delete the entire contents of an array.
- The use of **nextfile** to abandon processing of the current input file.
- The **and()**, **asort()**, **asorti()**, **bindtextdomain()**, **compl()**, **dcgettext()**, **dcngettext()**, **gensub()**, **lshift()**, **mktime()**, **or()**, **rshift()**, **strftime()**, **strtonum()**, **systime()** and **xor()** functions.
- Localizable strings.
- Adding new built-in functions dynamically with the **extension()** function.

The AWK book does not define the return value of the **close()** function. *Gawk*'s **close()** returns the value from *fclose(3)*, or *pclose(3)*, when closing an output file or pipe, respectively. It returns the process's exit status when closing an input pipe. The return value is -1 if the named file, pipe or co-process was not opened with a redirection.

When *gawk* is invoked with the **--traditional** option, if the *fs* argument to the **-F** option is "t", then **FS** is set to the tab character. Note that typing **gawk -Ft ...** simply causes the shell to quote the "t," and does not pass "t" to the **-F** option. Since this is a rather ugly special case, it is not the default behavior. This behavior also does not occur if **--posix** has been specified. To really get a tab character as the field separator, it is best to use single quotes: **gawk -F't' ...**

If *gawk* is *configured* with the **--enable-switch** option to the *configure* command, then it accepts an additional control-flow statement:

```
switch (expression) {
  case value|regex : statement
  ...
  [ default: statement ]
}
```

If *gawk* is *configured* with the **--disable-directories-fatal** option, then it will silently skip directories named on the command line. Otherwise, it will do so only if invoked with the **--traditional** option.

ENVIRONMENT VARIABLES

The **AWKPATH** environment variable can be used to provide a list of directories that *gawk* searches when looking for files named via the **-f** and **--file** options.

If **POSIXLY_CORRECT** exists in the environment, then *gawk* behaves exactly as if **--posix** had been specified on the command line. If **--lint** has been specified, *gawk* issues a warning message to this effect.

SEE ALSO

egrep(1), *getpid(2)*, *getppid(2)*, *getpgrp(2)*, *getuid(2)*, *geteuid(2)*, *getgid(2)*, *getegid(2)*, *getgroups(2)*

The AWK Programming Language, Alfred V. Aho, Brian W. Kernighan, Peter J. Weinberger, Addison-Wesley, 1988. ISBN 0-201-07981-X.

GAWK: Effective AWK Programming, Edition 3.0, published by the Free Software Foundation, 2001. The

current version of this document is available online at <http://www.gnu.org/software/gawk/manual>.

BUGS

The **-F** option is not necessary given the command line variable assignment feature; it remains only for backwards compatibility.

Syntactically invalid single character programs tend to overflow the parse stack, generating a rather unhelpful message. Such programs are surprisingly difficult to diagnose in the completely general case, and the effort to do so really is not worth it.

AUTHORS

The original version of UNIX *awk* was designed and implemented by Alfred Aho, Peter Weinberger, and Brian Kernighan of Bell Laboratories. Brian Kernighan continues to maintain and enhance it.

Paul Rubin and Jay Fenlason, of the Free Software Foundation, wrote *gawk*, to be compatible with the original version of *awk* distributed in Seventh Edition UNIX. John Woods contributed a number of bug fixes. David Trueman, with contributions from Arnold Robbins, made *gawk* compatible with the new version of UNIX *awk*. Arnold Robbins is the current maintainer.

The initial DOS port was done by Conrad Kwok and Scott Garfinkle. Scott Deifik is the current DOS maintainer. Pat Rankin did the port to VMS, and Michal Jaegermann did the port to the Atari ST. The port to OS/2 was done by Kai Uwe Rommel, with contributions and help from Darrel Hankerson. Juan M. Guerrero now maintains the OS/2 port. Fred Fish supplied support for the Amiga, and Martin Brown provided the BeOS port. Stephen Davies provided the original Tandem port, and Matthew Woehlke provided changes for Tandem's POSIX-compliant systems.

VERSION INFORMATION

This man page documents *gawk*, version 3.1.6.

BUG REPORTS

If you find a bug in *gawk*, please send electronic mail to bug-gawk@gnu.org. Please include your operating system and its revision, the version of *gawk* (from **gawk --version**), what C compiler you used to compile it, and a test program and data that are as small as possible for reproducing the problem.

Before sending a bug report, please do the following things. First, verify that you have the latest version of *gawk*. Many bugs (usually subtle ones) are fixed at each release, and if yours is out of date, the problem may already have been solved. Second, please see if setting the environment variable **LC_ALL** to **LC_ALL=C** causes things to behave as you expect. If so, it's a locale issue, and may or may not really be a bug. Finally, please read this man page and the reference manual carefully to be sure that what you think is a bug really is, instead of just a quirk in the language.

Whatever you do, do **NOT** post a bug report in **comp.lang.awk**. While the *gawk* developers occasionally read this newsgroup, posting bug reports there is an unreliable way to report bugs. Instead, please use the electronic mail addresses given above.

If you're using a GNU/Linux system or BSD-based system, you may wish to submit a bug report to the vendor of your distribution. That's fine, but please send a copy to the official email address as well, since there's no guarantee that the bug will be forwarded to the *gawk* maintainer.

ACKNOWLEDGEMENTS

Brian Kernighan of Bell Laboratories provided valuable assistance during testing and debugging. We thank him.

COPYING PERMISSIONS

Copyright © 1989, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2001, 2002, 2003, 2004, 2005, 2007 Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual page provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual page under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual page into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

NAME

gecos – return the GECOS (comment) entry for a user account

SYNOPSIS

edrc/lib/gecos [**-h**]

gecos *username*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the GECOS field (=comment) of a given user.

If the user does not exist on the system (or in an equivalent coming from some server somewhere) an empty string is returned.

OPTIONS

-h print usage message.

username name of an existing user on the system.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 user does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **gid(3)**, **group(3)**, **homedir(3)**, **uid(3)**

NOTES

-

BUGS

-

AUTHOR

gecos was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

getfilesystem – print the filesystem of the current working directory

SYNOPSIS

edrc/lib/getfilesystem

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **getfilesystem** the filesystem (special file) is printed where the current working directory is located on.

This output is basically identical to:

```
df -k . | awk 'END{print $1}'      # Solaris, Linux
bdf . | awk 'END{print $1}'        # HP-UX
xbdf . | awk 'END{print $1}'       # all
```

but is much faster especially if there are large file systems on HP-UX.

getfilesystem is used internally by **pkg_hostname** where fast filesystem resolving is important.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|--------------------------------------|
| 0 | filesystem could be resolved |
| 1 | error while resolving the filesystem |

FILES

/etc/mnttab on HP-UX and Solaris.

/etc/mntab on Linux.

EXAMPLES

-

SEE ALSO

edrcintro(1), **getmountpoint(3)**, **pkg_hostname(3)**, **bdf(1)**, **df(1)**, **xbdf(1)**

NOTES

The program is based on the source code in **compat.c** which is part of a **df** implementation written by Jeff Forys <jeffware@marjum.com>.

BUGS

-

AUTHOR

getfilesystem was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

gethostbyname – host lookup using gethostbyname() system call

SYNOPSIS

edrc/bin/gethostbyname

AVAILABILITY

WA2L/edrc

DESCRIPTION

gethostbyname is an interactive command using only the gethostbyname system call to resolve the host-name(s) and IP address(s). This ensures that the settings in the **/etc/nsswitch.conf** file are processed as the system does it. **gethostbyname** does not super-set the system settings in any kind like other resolving tools sometimes do.

To exit **gethostbyname** type *exit* at the **gethostbyname** command prompt.

The source code of **gethostbyname** is available in **edrc/src/gethostbyname.c**.

OPTIONS

-

OUTPUT INTERPRETATION

The output fields correspond to the *hostent* structure as defined in **netdb.h**.

h_error If the name server is not being used, the value of **h_errno** may not be meaningful.

1: **HOST_NOT_FOUND**

Authoritative answer host not found.

This value is returned on successful queries (what is a bit weird, of course).

2: **TRY_AGAIN**

Non-Authoritative Host not found, or SERVERFAIL.

This is usually a temporary error. The local server did not receive a response from an authoritative server. A retry at some later time may succeed. Or the queried host simply does not exist.

3: **NO_RECOVERY**

Non recoverable errors, FORMERR, REFUSED, NOTIMP.

This is a non-recoverable error.

4: NO_DATA

Valid name, no data record of requested type.

4: NO_ADDRESS

no address, look for MX record. The requested name is valid but does not have an IP address; this is not a temporary error. This means another type of request to the name server will result in an answer.

h_name The official name of the host.

h_aliases An array of alternate names for the host.

h_addrtype

The type of address being returned; always *AF_INET*.

h_length The length, in bytes, of the address.

h_addr_list

An array of network addresses for the host

EXIT STATUS

0 always

SEE ALSO

edrcintro(1), **gethostbyname(3N)**, **gethostent(3N)**, **nsswitch.conf(4)**, **resolve.conf(4)**, **resolver(4)**

BUGS

IPv6 is not supported yet.

AUTHOR

gethostbyname was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

getmountpoint – print the mountpoint of the current working directory

SYNOPSIS

edrc/lib/getmountpoint

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **getmountpoint** the mount point is printed where the current working directory is located on.

This output is basically identical to:

```
df -k . | awk 'END{print $6}'      # Solaris, Linux
bdf . | awk 'END{print $6}'        # HP-UX
xbdf . | awk 'END{print $6}'       # all
```

but is much faster especially if there are large file systems on HP-UX.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---------------------------------------|
| 0 | mount point could be resolved |
| 1 | error while resolving the mount point |

FILES

/etc/mnttab on HP-UX and Solaris.

/etc/mstab on Linux.

EXAMPLES

-

SEE ALSO

edrcintro(1), **getfilesystem(3)**, **bdf(1)**, **df(1)**, **xbdf(1)**

NOTES

The program is based on the source code in **compat.c** which is part of a **df** implementation written by Jeff Forys <jeffware@marjum.com>.

BUGS

-

AUTHOR

getmountpoint was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

gid – return the numeric group-id (GID) of a group

SYNOPSIS

edrc/lib/gid [**-h**]

gid *groupname*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the numeric group-id (GID) of a given group.

If the group does not exist on the system (or in an equivalent coming from some server somewhere) an empty string is returned.

OPTIONS

-h print usage message.

groupname
name of an existing group on the system.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 group does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **gecos(3)**, **group(3)**, **homedir(3)**, **uid(3)**, **user(3)**

NOTES

-

BUGS

-

AUTHOR

gid was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

glibc.version – print GLIBC version

SYNOPSIS

edrc/lib/glibc.version

edrc/lib/<OSID>/glibc.version

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the GLIBC version available on the system.

OPTIONS

-

EXIT STATUS

0 no error.

106 operating system (OSID) not supported.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ldd(1)**, **osid(3)**, **osid.probe(3)**

NOTES

Normally the GLIBC version is identical to the version of the **ldd** command.

BUGS

-

AUTHOR

glibc.version was developed by Christian Walther. Send suggestions and bug reports to wa21@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

grep, **egrep**, **fgrep**, **rgrep** – print lines matching a pattern

SYNOPSIS

```
grep [OPTIONS] PATTERN [FILE...]
grep [OPTIONS] -e PATTERN ... [FILE...]
grep [OPTIONS] -f FILE ... [FILE...]
```

DESCRIPTION

grep searches for *PATTERN* in each *FILE*. A *FILE* of “-” stands for standard input. If no *FILE* is given, recursive searches examine the working directory, and nonrecursive searches read standard input. By default, **grep** prints the matching lines.

In addition, the variant programs **egrep**, **fgrep** and **rgrep** are the same as **grep -E**, **grep -F**, and **grep -r**, respectively. These variants are deprecated, but are provided for backward compatibility.

OPTIONS**Generic Program Information**

--help Output a usage message and exit.

-V, --version
Output the version number of **grep** and exit.

Matcher Selection

-E, --extended-regexp
Interpret *PATTERN* as an extended regular expression (ERE, see below).

-F, --fixed-strings
Interpret *PATTERN* as a list of fixed strings (instead of regular expressions), separated by newlines, any of which is to be matched.

-G, --basic-regexp
Interpret *PATTERN* as a basic regular expression (BRE, see below). This is the default.

-P, --perl-regexp
Interpret the pattern as a Perl-compatible regular expression (PCRE). This is experimental and **grep -P** may warn of unimplemented features.

Matching Control

-e PATTERN, --regexp=PATTERN
Use *PATTERN* as the pattern. If this option is used multiple times or is combined with the **-f** (**--file**) option, search for all patterns given. This option can be used to protect a pattern beginning with “-”.

-f FILE, --file=FILE
Obtain patterns from *FILE*, one per line. If this option is used multiple times or is combined with the **-e** (**--regexp**) option, search for all patterns given. The empty file contains zero patterns, and therefore matches nothing.

-i, --ignore-case
Ignore case distinctions, so that characters that differ only in case match each other.

-v, --invert-match
Invert the sense of matching, to select non-matching lines.

-w, --word-regexp
Select only those lines containing matches that form whole words. The test is that the matching substring must either be at the beginning of the line, or preceded by a non-word constituent character. Similarly, it must be either at the end of the line or followed by a non-word constituent character. Word-constituent characters are letters, digits, and the underscore. This option has no effect if **-x** is also specified.

-x, --line-regexp

Select only those matches that exactly match the whole line. For a regular expression pattern, this is like parenthesizing the pattern and then surrounding it with `^` and `$`.

-y Obsolete synonym for **-i**.**General Output Control****-c, --count**

Suppress normal output; instead print a count of matching lines for each input file. With the **-v**, **--invert-match** option (see below), count non-matching lines.

--color[=WHEN], --colour[=WHEN]

Surround the matched (non-empty) strings, matching lines, context lines, file names, line numbers, byte offsets, and separators (for fields and groups of context lines) with escape sequences to display them in color on the terminal. The colors are defined by the environment variable **GREP_COLORS**. The deprecated environment variable **GREP_COLOR** is still supported, but its setting does not have priority. *WHEN* is **never**, **always**, or **auto**.

-L, --files-without-match

Suppress normal output; instead print the name of each input file from which no output would normally have been printed. The scanning will stop on the first match.

-l, --files-with-matches

Suppress normal output; instead print the name of each input file from which output would normally have been printed. The scanning will stop on the first match.

-m NUM, --max-count=NUM

Stop reading a file after *NUM* matching lines. If the input is standard input from a regular file, and *NUM* matching lines are output, **grep** ensures that the standard input is positioned to just after the last matching line before exiting, regardless of the presence of trailing context lines. This enables a calling process to resume a search. When **grep** stops after *NUM* matching lines, it outputs any trailing context lines. When the **-c** or **--count** option is also used, **grep** does not output a count greater than *NUM*. When the **-v** or **--invert-match** option is also used, **grep** stops after outputting *NUM* non-matching lines.

-o, --only-matching

Print only the matched (non-empty) parts of a matching line, with each such part on a separate output line.

-q, --quiet, --silent

Quiet; do not write anything to standard output. Exit immediately with zero status if any match is found, even if an error was detected. Also see the **-s** or **--no-messages** option.

-s, --no-messages

Suppress error messages about nonexistent or unreadable files.

Output Line Prefix Control**-b, --byte-offset**

Print the 0-based byte offset within the input file before each line of output. If **-o** (**--only-matching**) is specified, print the offset of the matching part itself.

-H, --with-filename

Print the file name for each match. This is the default when there is more than one file to search.

-h, --no-filename

Suppress the prefixing of file names on output. This is the default when there is only one file (or only standard input) to search.

--label=LABEL

Display input actually coming from standard input as input coming from file *LABEL*. This is especially useful when implementing tools like **zgrep**, e.g., **gzip -cd foo.gz | grep --label=foo -H something**. See also the **-H** option.

-n, --line-number

Prefix each line of output with the 1-based line number within its input file.

-T, --initial-tab

Make sure that the first character of actual line content lies on a tab stop, so that the alignment of tabs looks normal. This is useful with options that prefix their output to the actual content: **-H**, **-n**, and **-b**. In order to improve the probability that lines from a single file will all start at the same column, this also causes the line number and byte offset (if present) to be printed in a minimum size field width.

-u, --unix-byte-offsets

Report Unix-style byte offsets. This switch causes **grep** to report byte offsets as if the file were a Unix-style text file, i.e., with CR characters stripped off. This will produce results identical to running **grep** on a Unix machine. This option has no effect unless **-b** option is also used; it has no effect on platforms other than MS-DOS and MS-Windows.

-Z, --null

Output a zero byte (the ASCII NUL character) instead of the character that normally follows a file name. For example, **grep -IZ** outputs a zero byte after each file name instead of the usual newline. This option makes the output unambiguous, even in the presence of file names containing unusual characters like newlines. This option can be used with commands like **find -print0**, **perl -0**, **sort -z**, and **xargs -0** to process arbitrary file names, even those that contain newline characters.

Context Line Control**-A NUM, --after-context=NUM**

Print *NUM* lines of trailing context after matching lines. Places a line containing a group separator (--) between contiguous groups of matches. With the **-o** or **--only-matching** option, this has no effect and a warning is given.

-B NUM, --before-context=NUM

Print *NUM* lines of leading context before matching lines. Places a line containing a group separator (--) between contiguous groups of matches. With the **-o** or **--only-matching** option, this has no effect and a warning is given.

-C NUM, -NUM, --context=NUM

Print *NUM* lines of output context. Places a line containing a group separator (--) between contiguous groups of matches. With the **-o** or **--only-matching** option, this has no effect and a warning is given.

File and Directory Selection**-a, --text**

Process a binary file as if it were text; this is equivalent to the **--binary-files=text** option.

--binary-files=TYPE

If a file's data or metadata indicate that the file contains binary data, assume that the file is of type *TYPE*. Non-text bytes indicate binary data; these are either output bytes that are improperly encoded for the current locale, or null input bytes when the **-z** option is not given.

By default, *TYPE* is **binary**, and when **grep** discovers that a file is binary it suppresses any further output, and instead outputs either a one-line message saying that a binary file matches, or no message if there is no match.

If *TYPE* is **without-match**, when **grep** discovers that a file is binary it assumes that the rest of the file does not match; this is equivalent to the **-I** option.

If *TYPE* is **text**, **grep** processes a binary file as if it were text; this is equivalent to the **-a** option.

When *type* is **binary**, **grep** may treat non-text bytes as line terminators even without the **-z** option. This means choosing **binary** versus **text** can affect whether a pattern matches a file. For example, when *type* is **binary** the pattern **q\$** might match **q** immediately followed by a null byte, even though this is not matched when *type* is **text**. Conversely, when *type* is **binary** the pattern **.**

(period) might not match a null byte.

Warning: The **-a** option might output binary garbage, which can have nasty side effects if the output is a terminal and if the terminal driver interprets some of it as commands. On the other hand, when reading files whose text encodings are unknown, it can be helpful to use **-a** or to set **LC_ALL='C'** in the environment, in order to find more matches even if the matches are unsafe for direct display.

-D ACTION, --devices=ACTION

If an input file is a device, FIFO or socket, use *ACTION* to process it. By default, *ACTION* is **read**, which means that devices are read just as if they were ordinary files. If *ACTION* is **skip**, devices are silently skipped.

-d ACTION, --directories=ACTION

If an input file is a directory, use *ACTION* to process it. By default, *ACTION* is **read**, i.e., read directories just as if they were ordinary files. If *ACTION* is **skip**, silently skip directories. If *ACTION* is **recurse**, read all files under each directory, recursively, following symbolic links only if they are on the command line. This is equivalent to the **-r** option.

--exclude=GLOB

Skip any command-line file with a name suffix that matches the pattern *GLOB*, using wildcard matching; a name suffix is either the whole name, or any suffix starting after a */* and before a *+non-/. When searching recursively, skip any subfile whose base name matches *GLOB*; the base name is the part after the last */*. A pattern can use ***, *?*, and *[...]* as wildcards, and ** to quote a wildcard or backslash character literally.*

--exclude-from=FILE

Skip files whose base name matches any of the file-name globs read from *FILE* (using wildcard matching as described under **--exclude**).

--exclude-dir=GLOB

Skip any command-line directory with a name suffix that matches the pattern *GLOB*. When searching recursively, skip any subdirectory whose base name matches *GLOB*. Ignore any redundant trailing slashes in *GLOB*.

-I Process a binary file as if it did not contain matching data; this is equivalent to the **--binary-files=without-match** option.

--include=GLOB

Search only files whose base name matches *GLOB* (using wildcard matching as described under **--exclude**).

-r, --recursive

Read all files under each directory, recursively, following symbolic links only if they are on the command line. Note that if no file operand is given, *grep* searches the working directory. This is equivalent to the **-d recurse** option.

-R, --dereference-recursive

Read all files under each directory, recursively. Follow all symbolic links, unlike **-r**.

Other Options

--line-buffered

Use line buffering on output. This can cause a performance penalty.

-U, --binary

Treat the file(s) as binary. By default, under MS-DOS and MS-Windows, **grep** guesses whether a file is text or binary as described for the **--binary-files** option. If **grep** decides the file is a text file, it strips the CR characters from the original file contents (to make regular expressions with *^* and *\$* work correctly). Specifying **-U** overrules this guesswork, causing all files to be read and passed to the matching mechanism verbatim; if the file is a text file with CR/LF pairs at the end of each line, this will cause some regular expressions to fail. This option has no effect on platforms other than MS-DOS and MS-Windows.

-z, --null-data

Treat input and output data as sequences of lines, each terminated by a zero byte (the ASCII NUL character) instead of a newline. Like the **-Z** or **--null** option, this option can be used with commands like **sort -z** to process arbitrary file names.

REGULAR EXPRESSIONS

A regular expression is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions.

grep understands three different versions of regular expression syntax: “basic” (BRE), “extended” (ERE) and “perl” (PCRE). In GNU **grep** there is no difference in available functionality between basic and extended syntaxes. In other implementations, basic regular expressions are less powerful. The following description applies to extended regular expressions; differences for basic regular expressions are summarized afterwards. Perl-compatible regular expressions give additional functionality, and are documented in `pcresyntax(3)` and `pcrpattern(3)`, but work only if PCRE is available in the system.

The fundamental building blocks are the regular expressions that match a single character. Most characters, including all letters and digits, are regular expressions that match themselves. Any meta-character with special meaning may be quoted by preceding it with a backslash.

The period `.` matches any single character.

Character Classes and Bracket Expressions

A *bracket expression* is a list of characters enclosed by `[` and `]`. It matches any single character in that list; if the first character of the list is the caret `^` then it matches any character *not* in the list. For example, the regular expression `[0123456789]` matches any single digit.

Within a bracket expression, a *range expression* consists of two characters separated by a hyphen. It matches any single character that sorts between the two characters, inclusive, using the locale’s collating sequence and character set. For example, in the default C locale, `[a-d]` is equivalent to `[abcd]`. Many locales sort characters in dictionary order, and in these locales `[a-d]` is typically not equivalent to `[abcd]`; it might be equivalent to `[aBbCcDd]`, for example. To obtain the traditional interpretation of bracket expressions, you can use the C locale by setting the `LC_ALL` environment variable to the value `C`.

Finally, certain named classes of characters are predefined within bracket expressions, as follows. Their names are self explanatory, and they are `[:alnum:]`, `[:alpha:]`, `[:cntrl:]`, `[:digit:]`, `[:graph:]`, `[:lower:]`, `[:print:]`, `[:punct:]`, `[:space:]`, `[:upper:]`, and `[:xdigit:]`. For example, `[:alnum:]` means the character class of numbers and letters in the current locale. In the C locale and ASCII character set encoding, this is the same as `[0-9A-Za-z]`. (Note that the brackets in these class names are part of the symbolic names, and must be included in addition to the brackets delimiting the bracket expression.) Most meta-characters lose their special meaning inside bracket expressions. To include a literal `]` place it first in the list. Similarly, to include a literal `^` place it anywhere but first. Finally, to include a literal `-` place it last.

Anchoring

The caret `^` and the dollar sign `$` are meta-characters that respectively match the empty string at the beginning and end of a line.

The Backslash Character and Special Expressions

The symbols `\<` and `\>` respectively match the empty string at the beginning and end of a word. The symbol `\b` matches the empty string at the edge of a word, and `\B` matches the empty string provided it’s *not* at the edge of a word. The symbol `\w` is a synonym for `[_[:alnum:]]` and `\W` is a synonym for `[^_[:alnum:]]`.

Repetition

A regular expression may be followed by one of several repetition operators:

- ?** The preceding item is optional and matched at most once.
- *** The preceding item will be matched zero or more times.
- +** The preceding item will be matched one or more times.
- {*n*}** The preceding item is matched exactly *n* times.
- {*n*,}** The preceding item is matched *n* or more times.

- {,m}** The preceding item is matched at most *m* times. This is a GNU extension.
- {n,m}** The preceding item is matched at least *n* times, but not more than *m* times.

Concatenation

Two regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated expressions.

Alternation

Two regular expressions may be joined by the infix operator `|`; the resulting regular expression matches any string matching either alternate expression.

Precedence

Repetition takes precedence over concatenation, which in turn takes precedence over alternation. A whole expression may be enclosed in parentheses to override these precedence rules and form a subexpression.

Back References and Subexpressions

The back-reference `\n`, where *n* is a single digit, matches the substring previously matched by the *n*th parenthesized subexpression of the regular expression.

Basic vs Extended Regular Expressions

In basic regular expressions the meta-characters `?`, `+`, `{`, `|`, `(`, and `)` lose their special meaning; instead use the backslashed versions `\?`, `\+`, `\{`, `\|`, `\(`, and `\)`.

ENVIRONMENT VARIABLES

The behavior of **grep** is affected by the following environment variables.

The locale for category **LC_foo** is specified by examining the three environment variables **LC_ALL**, **LC_foo**, **LANG**, in that order. The first of these variables that is set specifies the locale. For example, if **LC_ALL** is not set, but **LC_MESSAGES** is set to **pt_BR**, then the Brazilian Portuguese locale is used for the **LC_MESSAGES** category. The C locale is used if none of these environment variables are set, if the locale catalog is not installed, or if **grep** was not compiled with national language support (NLS). The shell command **locale -a** lists locales that are currently available.

GREP_OPTIONS

This variable specifies default options to be placed in front of any explicit options. As this causes problems when writing portable scripts, this feature will be removed in a future release of **grep**, and **grep** warns if it is used. Please use an alias or script instead.

GREP_COLOR

This variable specifies the color used to highlight matched (non-empty) text. It is deprecated in favor of **GREP_COLORS**, but still supported. The **mt**, **ms**, and **mc** capabilities of **GREP_COLORS** have priority over it. It can only specify the color used to highlight the matching non-empty text in any matching line (a selected line when the **-v** command-line option is omitted, or a context line when **-v** is specified). The default is **01;31**, which means a bold red foreground text on the terminal's default background.

GREP_COLORS

Specifies the colors and other attributes used to highlight various parts of the output. Its value is a colon-separated list of capabilities that defaults to **ms=01;31:mc=01;31:sl=:cx=:fn=35:ln=32:bn=32:se=36** with the **rv** and **ne** boolean capabilities omitted (i.e., false). Supported capabilities are as follows.

- sl=** SGR substring for whole selected lines (i.e., matching lines when the **-v** command-line option is omitted, or non-matching lines when **-v** is specified). If however the boolean **rv** capability and the **-v** command-line option are both specified, it applies to context matching lines instead. The default is empty (i.e., the terminal's default color pair).
- cx=** SGR substring for whole context lines (i.e., non-matching lines when the **-v** command-line option is omitted, or matching lines when **-v** is specified). If however the boolean **rv** capability and the **-v** command-line option are both specified, it applies to selected non-matching lines instead. The default is empty (i.e., the terminal's default color pair).

- rv** Boolean value that reverses (swaps) the meanings of the **sl=** and **cx=** capabilities when the **-v** command-line option is specified. The default is false (i.e., the capability is omitted).
- mt=01;31** SGR substring for matching non-empty text in any matching line (i.e., a selected line when the **-v** command-line option is omitted, or a context line when **-v** is specified). Setting this is equivalent to setting both **ms=** and **mc=** at once to the same value. The default is a bold red text foreground over the current line background.
- ms=01;31** SGR substring for matching non-empty text in a selected line. (This is only used when the **-v** command-line option is omitted.) The effect of the **sl=** (or **cx=** if **rv**) capability remains active when this kicks in. The default is a bold red text foreground over the current line background.
- mc=01;31** SGR substring for matching non-empty text in a context line. (This is only used when the **-v** command-line option is specified.) The effect of the **cx=** (or **sl=** if **rv**) capability remains active when this kicks in. The default is a bold red text foreground over the current line background.
- fn=35** SGR substring for file names prefixing any content line. The default is a magenta text foreground over the terminal's default background.
- ln=32** SGR substring for line numbers prefixing any content line. The default is a green text foreground over the terminal's default background.
- bn=32** SGR substring for byte offsets prefixing any content line. The default is a green text foreground over the terminal's default background.
- se=36** SGR substring for separators that are inserted between selected line fields (:), between context line fields, (–), and between groups of adjacent lines when nonzero context is specified (––). The default is a cyan text foreground over the terminal's default background.
- ne** Boolean value that prevents clearing to the end of line using Erase in Line (EL) to Right (**\33[K**) each time a colorized item ends. This is needed on terminals on which EL is not supported. It is otherwise useful on terminals for which the **back_color_erase** (**bce**) boolean terminfo capability does not apply, when the chosen highlight colors do not affect the background, or when EL is too slow or causes too much flicker. The default is false (i.e., the capability is omitted).

Note that boolean capabilities have no **=...** part. They are omitted (i.e., false) by default and become true when specified.

See the Select Graphic Rendition (SGR) section in the documentation of the text terminal that is used for permitted values and their meaning as character attributes. These substring values are integers in decimal representation and can be concatenated with semicolons. **grep** takes care of assembling the result into a complete SGR sequence (**\33[...m**). Common values to concatenate include **1** for bold, **4** for underline, **5** for blink, **7** for inverse, **39** for default foreground color, **30** to **37** for foreground colors, **90** to **97** for 16-color mode foreground colors, **38;5;0** to **38;5;255** for 88-color and 256-color modes foreground colors, **49** for default background color, **40** to **47** for background colors, **100** to **107** for 16-color mode background colors, and **48;5;0** to **48;5;255** for 88-color and 256-color modes background colors.

LC_ALL, LC_COLLATE, LANG

These variables specify the locale for the **LC_COLLATE** category, which determines the collating sequence used to interpret range expressions like **[a–z]**.

LC_ALL, LC_CTYPE, LANG

These variables specify the locale for the **LC_CTYPE** category, which determines the type of characters, e.g., which characters are whitespace. This category also determines the character encoding, that is, whether text is encoded in UTF-8, ASCII, or some other encoding. In the C or POSIX locale, all characters are encoded as a single byte and every byte is a valid character.

LC_ALL, LC_MESSAGES, LANG

These variables specify the locale for the **LC_MESSAGES** category, which determines the language that **grep** uses for messages. The default C locale uses American English messages.

POSIXLY_CORRECT

If set, **grep** behaves as POSIX requires; otherwise, **grep** behaves more like other GNU programs. POSIX requires that options that follow file names must be treated as file names; by default, such options are permuted to the front of the operand list and are treated as options. Also, POSIX requires that unrecognized options be diagnosed as “illegal”, but since they are not really against the law the default is to diagnose them as “invalid”. **POSIXLY_CORRECT** also disables **_GNU_nonoption_argv_flags_**, described below.

_GNU_nonoption_argv_flags_

(Here *N* is **grep**’s numeric process ID.) If the *i*th character of this environment variable’s value is **1**, do not consider the *i*th operand of **grep** to be an option, even if it appears to be one. A shell can put this variable in the environment for each command it runs, specifying which operands are the results of file name wildcard expansion and therefore should not be treated as options. This behavior is available only with the GNU C library, and only when **POSIXLY_CORRECT** is not set.

EXIT STATUS

Normally the exit status is 0 if a line is selected, 1 if no lines were selected, and 2 if an error occurred. However, if the **-q** or **--quiet** or **--silent** is used and a line is selected, the exit status is 0 even if an error occurred.

COPYRIGHT

Copyright 1998–2000, 2002, 2005–2017 Free Software Foundation, Inc.

This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

BUGS**Reporting Bugs**

Email bug reports to [the bug-reporting address](mailto:bug-grep@gnu.org) (bug-grep@gnu.org). An [email archive](http://lists.gnu.org/mailman/listinfo/bug-grep) (http://lists.gnu.org/mailman/listinfo/bug-grep) and a [bug tracker](http://debbugs.gnu.org/cgi/pkgreport.cgi?package=grep) (http://debbugs.gnu.org/cgi/pkgreport.cgi?package=grep) are available.

Known Bugs

Large repetition counts in the $\{n,m\}$ construct may cause **grep** to use lots of memory. In addition, certain other obscure regular expressions require exponential time and space, and may cause **grep** to run out of memory.

Back-references are very slow, and may require exponential time.

SEE ALSO**Regular Manual Pages**

awk(1), cmp(1), diff(1), find(1), gzip(1), perl(1), sed(1), sort(1), xargs(1), zgrep(1), read(2), pcre(3), pcresyntax(3), pcrepattern(3), terminfo(5), glob(7), regex(7).

POSIX Programmer’s Manual Page

grep(1p).

Full Documentation

A [complete manual](http://www.gnu.org/software/grep/manual/) (http://www.gnu.org/software/grep/manual/) is available. If the **info** and **grep** programs are properly installed at your site, the command

info grep

should give you access to the complete manual.

NOTES

This man page is maintained only fitfully; the full documentation is often more up-to-date.

NAME

group – return the group name for a numeric group-id (gid)

SYNOPSIS

edrc/lib/group [**-h**]

group *group-id*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the group name for a given numeric group-id (gid).

If the group does not exist on the system (or in an equivalent coming from some server somewhere) an empty string is returned.

OPTIONS

-h print usage message.

group-id numeric group id of an existing group on the system.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 group for given *group-id* does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **gecos(3)**, **gid(3)**, **homedir(3)**, **uid(3)**, **user(3)**

NOTES

-

BUGS

-

AUTHOR

group was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

gs – Ghostscript (PostScript and PDF language interpreter and previewer)

SYNOPSIS

gs [*options*] [*files*] ...

DESCRIPTION

The **gs** command invokes **Ghostscript**, an interpreter of Adobe Systems' **PostScript**(tm) and **Portable Document Format** (PDF) languages. **gs** reads "files" in sequence and executes them as Ghostscript programs. After doing this, it reads further input from the standard input stream (normally the keyboard), interpreting each line separately and output to an output device (may be a file or an X11 window preview, see below). The interpreter exits gracefully when it encounters the "quit" command (either in a file or from the keyboard), at end-of-file, or at an interrupt signal (such as Control-C at the keyboard).

The interpreter recognizes many option switches, some of which are described below. Please see the usage documentation for complete information. Switches may appear anywhere in the command line and apply to all files thereafter. Invoking Ghostscript with the **-h** or **-?** switch produces a message which shows several useful switches, all the devices known to that executable, and the search path for fonts; on Unix it also shows the location of detailed documentation.

Ghostscript may be built to use many different output devices. To see which devices your executable includes, run "**gs -h**".

Unless you specify a particular device, Ghostscript normally opens the first one of those and directs output to it.

If you have installed the ghostscript-x Debian package and are under X, the default device is an X11 window (previewer), else ghostscript will use the bbox device and print on stdout the dimension of the postscript file.

So if the first one in the list is the one you want to use, just issue the command

```
gs myfile.ps
```

You can also check the set of available devices from within Ghostscript: invoke Ghostscript and type

```
devicenames ==
```

but the first device on the resulting list may not be the default device you determine with "**gs -h**". To specify "AbcXyz" as the initial output device, include the switch

```
-sDEVICE=AbcXyz
```

For example, for output to an Epson printer you might use the command

```
gs -sDEVICE=epson myfile.ps
```

The "**-sDEVICE=**" switch must precede the first mention of a file to print, and only the switch's first use has any effect.

Finally, you can specify a default device in the environment variable **GS_DEVICE**. The order of precedence for these alternatives from highest to lowest (Ghostscript uses the device defined highest in the list) is:

Some devices can support different resolutions (densities). To specify the resolution on such a printer, use the "**-r**" switch:

```
gs -sDEVICE=<device> -r<xres>x<yres>
```

For example, on a 9-pin Epson-compatible printer, you get the lowest-density (fastest) mode with

```
gs -sDEVICE=epson -r60x72
```

and the highest-density (best output quality) mode with

```
gs -sDEVICE=epson -r240x72.
```

If you select a printer as the output device, Ghostscript also allows you to choose where Ghostscript sends

the output — on Unix systems, usually to a temporary file. To send the output to a file "foo.xyz", use the switch

```
-sOutputFile=foo.xyz
```

You might want to print each page separately. To do this, send the output to a series of files "foo1.xyz, foo2.xyz, ..." using the "-sOutputFile=" switch with "%d" in a filename template:

```
-sOutputFile=foo%d.xyz
```

Each resulting file receives one page of output, and the files are numbered in sequence. "%d" is a printf format specification; you can also use a variant like "%02d".

You can also send output to a pipe. For example, to pipe output to the "**lpr**" command (which, on many Unix systems, directs it to a printer), use the option

```
-sOutputFile=%pipe%lpr
```

You can also send output to standard output:

```
-sOutputFile=--
```

or

```
-sOutputFile=%stdout%
```

In this case you must also use the **-q** switch, to prevent Ghostscript from writing messages to standard output.

To select a specific paper size, use the command line switch

```
-sPAPERSIZE=<paper_size>
```

for instance

```
-sPAPERSIZE=a4
```

or

```
-sPAPERSIZE=legal
```

Most ISO and US paper sizes are recognized. See the usage documentation for a full list, or the definitions in the initialization file "gs_statd.ps".

Ghostscript can do many things other than print or view PostScript and PDF files. For example, if you want to know the bounding box of a PostScript (or EPS) file, Ghostscript provides a special "device" that just prints out this information.

For example, using one of the example files distributed with Ghostscript,

```
gs -sDEVICE=bbbox golfer.ps
```

prints out

```
%%BoundingBox: 0 25 583 732
```

```
%%HiResBoundingBox: 0.808497 25.009496 582.994503 731.809445
```

OPTIONS

-- filename arg1 ...

Takes the next argument as a file name as usual, but takes all remaining arguments (even if they have the syntactic form of switches) and defines the name "ARGUMENTS" in "userdict" (not "systemdict") as an array of those strings, **before** running the file. When Ghostscript finishes executing the file, it exits back to the shell.

-Dname=token

-dname=token

Define a name in "systemdict" with the given definition. The token must be exactly one token (as defined by the "token" operator) and may contain no whitespace.

-Dname

- dname**
Define a name in "systemdict" with value=null.
- Sname=string**
-sname=string
Define a name in "systemdict" with a given string as value. This is different from **-d**. For example, **-dname=35** is equivalent to the program fragment
 /name 35 def
whereas **-sname=35** is equivalent to
 /name (35) def
- P**
Makes Ghostscript to look first in the current directory for library files. By default, Ghostscript no longer looks in the current directory, unless, of course, the first explicitly supplied directory is "." in **-I**. See also the **INITIALIZATION FILES** section below, and bundled **Use.htm** for detailed discussion on search paths and how Ghostscript finds files.
- q**
Quiet startup: suppress normal startup messages, and also do the equivalent of **-dQUIET**.
- gnumber1xnumber2**
Equivalent to **-dDEVICEWIDTH=number1** and **-dDEVICEHEIGHT=number2**. This is for the benefit of devices (such as X11 windows) that require (or allow) width and height to be specified.
- rnumber**
-rnumber1xnumber2
Equivalent to **-dDEVICEXRESOLUTION=number1** and **-dDEVICEYRESOLUTION=number2**. This is for the benefit of devices such as printers that support multiple X and Y resolutions. If only one number is given, it is used for both X and Y resolutions.
- Idirectories**
Adds the designated list of directories at the head of the search path for library files.
- This is not really a switch, but indicates to Ghostscript that standard input is coming from a file or a pipe and not interactively from the command line. Ghostscript reads from standard input until it reaches end-of-file, executing it like any other file, and then continues with processing the command line. When the command line has been entirely processed, Ghostscript exits rather than going into its interactive mode.

Note that the normal initialization file "gs_init.ps" makes "systemdict" read-only, so the values of names defined with **-D**, **-d**, **-S**, or **-s** cannot be changed (although, of course, they can be superseded by definitions in "userdict" or other dictionaries.)

SPECIAL NAMES

- dNOCACHE**
Disables character caching. Useful only for debugging.
- dNOBIND**
Disables the "bind" operator. Useful only for debugging.
- dNODISPLAY**
Suppresses the normal initialization of the output device. This may be useful when debugging.
- dNOPAUSE**
Disables the prompt and pause at the end of each page. This may be desirable for applications where another program is driving Ghostscript.
- dNOPLATFONTS**
Disables the use of fonts supplied by the underlying platform (for instance X Windows). This may be needed if the platform fonts look undesirably different from the scalable fonts.
- dSAFER**
Restricts file operations the job can perform. Strongly recommended for spoolers, conversion scripts or other sensitive environments where a badly written or malicious PostScript program

code must be prevented from changing important files.

-dWRITESYSTEMDICT

Leaves "systemdict" writable. This is necessary when running special utility programs, but is strongly discouraged as it bypasses normal Postscript security measures.

-sDEVICE=device

Selects an alternate initial output device, as described above.

-sOutputFile=filename

Selects an alternate output file (or pipe) for the initial output device, as described above.

SAFER MODE

The **-dSAFER** option disables the "deletefile" and "renamefile" operators and prohibits opening piped commands ("%pipe%cmd"). Only "%stdout" and "%stderr" can be opened for writing. It also disables reading from files, except for "%stdin", files given as a command line argument, and files contained in paths given by LIBPATH and FONTPATH or specified by the system params /FontResourceDir and /GenericResourceDir.

This mode also sets the .LockSafetyParams parameter of the initial output device to protect against programs that attempt to write to files using the OutputFile device parameter. Since the device parameters specified on the command line, including OutputFile, are set prior to SAFER mode, use of "-sOutputFile=..." on the command line is unrestricted.

SAFER mode prevents changing the /GenericResourceDir, /FontResourceDir, /SystemParamsPassword, and /StartJobPassword.

While SAFER mode is not the default, it is the default for many wrapper scripts such as ps2pdf and may be the default in a subsequent release of Ghostscript. Thus when running programs that need to open files or set restricted parameters you should pass the **-dNOSAFAFER** command line option or its synonym **-dDE-LAYSFAFER**.

When running with **-dNOSAFAFER** it is possible to perform a "save" followed by ".setsafe", execute a file or procedure in SAFER mode, and then use "restore" to return to NOSAFER mode. In order to prevent the save object from being restored by the foreign file or procedure, the ".runandhide" operator should be used to hide the save object from the restricted procedure.

FILES

The locations of many Ghostscript run-time files are compiled into the executable when it is built. Run "**gs -h**" to find the location of Ghostscript documentation on your system, from which you can get more details. On a Debian system they are in **/usr**.

/usr/share/ghostscript/[0-9]*.[0.9]*/*

Startup files, utilities, and basic font definitions (where [0-9]*.[0.9]* is the ghostscript version)

/usr/share/fonts/type1/gsfonts/*

More font definitions from the gsfonts package

/usr/share/doc/ghostscript/examples/*

Ghostscript demonstration files (if ghostscript-doc package is installed)

/usr/share/doc/ghostscript/*

Diverse document files (may need to install ghostscript-doc package)

INITIALIZATION FILES

When looking for the initialization files "gs_*.ps", the files related to fonts, or the file for the "run" operator, Ghostscript first tries to open the file with the name as given, using the current working directory if no directory is specified. If this fails, and the file name doesn't specify an explicit directory or drive (for instance, doesn't contain "/" on Unix systems), Ghostscript tries directories in this order:

1. the directories specified by the **-I** switches in the command line (see below), if any;
2. the directories specified by the **GS_LIB** environment variable, if any;

3. the directories specified by the **GS_LIB_DEFAULT** macro in the Ghostscript makefile when the executable was built. **GS_LIB_DEFAULT** is `"/usr/share/ghostscript/[0-9]*.[0-9]*/lib"` on a Debian system where `"[0-9]*.[0-9]*/"` represents the Ghostscript version number

Each of these (**GS_LIB_DEFAULT**, **GS_LIB**, and **-I** parameter) may be either a single directory or a list of directories separated by `":"`.

ENVIRONMENT

GS_OPTIONS

String of options to be processed before the command line options

GS_DEVICE

Used to specify an output device

GS_FONTPATH

Path names used to search for fonts

GS_LIB

Path names for initialization files and fonts

TEMP Where temporary files are made

X RESOURCES

Ghostscript, or more properly the X11 display device, looks for the following resources under the program name "Ghostscript":

borderWidth

The border width in pixels (default = 1).

borderColor

The name of the border color (default = black).

geometry

The window size and placement, `WxH+X+Y` (default is NULL).

xResolution

The number of x pixels per inch (default is computed from **WidthOfScreen** and **WidthMMOfScreen**).

yResolution

The number of y pixels per inch (default is computed from **HeightOfScreen** and **HeightMMOfScreen**).

useBackingPixmap

Determines whether backing store is to be used for saving display window (default = true).

See the usage document for a more complete list of resources. To set these resources on Unix, put them in a file such as `"~/Xresources"` in the following form:

```
Ghostscript*geometry: 612x792-0+0
Ghostscript*xResolution: 72
Ghostscript*yResolution: 72
```

Then merge these resources into the X server's resource database:

```
% xrdp -merge ~/.Xresources
```

SEE ALSO

edrcintro(1)

The various Ghostscript document files (above), especially **Use.htm**. On Debian you may need to install `ghostscript-doc` before reading the documentation.

BUGS

See <http://bugs.ghostscript.com/> and the Usenet news group comp.lang.postscript.

VERSION

This document was last revised for Ghostscript version 9.26.

AUTHOR

Artifex Software, Inc. are the primary maintainers of Ghostscript. Russell J. Lang, [gsview at ghostgum.com.au](mailto:gsview@ghostgum.com.au), is the author of most of the MS Windows code in Ghostscript.

NAME

gzip, *gunzip*, *zcat* – compress or expand files

SYNOPSIS

gzip [**-acdfhlLnNrtvV19**] [**-S suffix**] [*name ...*]

gunzip [**-acfhllNrtvV**] [**-S suffix**] [*name ...*]

zcat [**-fhLV**] [*name ...*]

DESCRIPTION

Gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension **.gz**, while keeping the same ownership modes, access and modification times. (The default extension is **-gz** for VMS, **z** for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. *Gzip* will only attempt to compress regular files. In particular, it will ignore symbolic links.

If the compressed file name is too long for its file system, *gzip* truncates it. *Gzip* attempts to truncate only the parts of the file name longer than 3 characters. (A part is delimited by dots.) If the name consists of small parts only, the longest parts are truncated. For example, if file names are limited to 14 characters, *gzip.msdos.exe* is compressed to *gzi.msd.exe.gz*. Names are not truncated on systems which do not have a limit on file name length.

By default, *gzip* keeps the original file name and timestamp in the compressed file. These are used when decompressing the file with the **-N** option. This is useful when the compressed file name was truncated or when the time stamp was not preserved after a file transfer.

Compressed files can be restored to their original form using *gzip -d* or *gunzip* or *zcat*. If the original name saved in the compressed file is not suitable for its file system, a new name is constructed from the original one to make it legal.

gunzip takes a list of files on its command line and replaces each file whose name ends with **.gz**, **-gz**, **.z**, **-z**, **_z** or **.Z** and which begins with the correct magic number with an uncompressed file without the original extension. *gunzip* also recognizes the special extensions **.tgz** and **.taz** as shorthands for **.tar.gz** and **.tar.Z** respectively. When compressing, *gzip* uses the **.tgz** extension if necessary instead of truncating a file with a **.tar** extension.

gunzip can currently decompress files created by *gzip*, *zip*, *compress*, *compress -H* or *pack*. The detection of the input format is automatic. When using the first two formats, *gunzip* checks a 32 bit CRC. For *pack*, *gunzip* checks the uncompressed length. The standard *compress* format was not designed to allow consistency checks. However *gunzip* is sometimes able to detect a bad **.Z** file. If you get an error when uncompressing a **.Z** file, do not assume that the **.Z** file is correct simply because the standard *uncompress* does not complain. This generally means that the standard *uncompress* does not check its input, and happily generates garbage output. The SCO *compress -H* format (lzh compression method) does not include a CRC but also allows some consistency checks.

Files created by *zip* can be uncompressed by *gzip* only if they have a single member compressed with the 'deflation' method. This feature is only intended to help conversion of *tar.zip* files to the *tar.gz* format. To extract a *zip* file with a single member, use a command like *gunzip <foo.zip* or *gunzip -S .zip foo.zip*. To extract *zip* files with several members, use *unzip* instead of *gunzip*.

zcat is identical to *gunzip -c*. (On some systems, *zcat* may be installed as *gzcat* to preserve the original link to *compress*.) *zcat* uncompresses either a list of files on the command line or its standard input and writes the uncompressed data on standard output. *zcat* will uncompress files that have the correct magic number whether they have a **.gz** suffix or not.

Gzip uses the Lempel-Ziv algorithm used in *zip* and PKZIP. The amount of compression obtained depends on the size of the input and the distribution of common substrings. Typically, text such as source code or English is reduced by 60–70%. Compression is generally much better than that achieved by LZW (as used in *compress*), Huffman coding (as used in *pack*), or adaptive Huffman coding (*compact*).

Compression is always performed, even if the compressed file is slightly larger than the original. The worst case expansion is a few bytes for the *gzip* file header, plus 5 bytes every 32K block, or an expansion ratio of

0.015% for large files. Note that the actual number of used disk blocks almost never increases. *gzip* preserves the mode, ownership and timestamps of files when compressing or decompressing.

The *gzip* file format is specified in P. Deutsch, GZIP file format specification version 4.3, <ftp://ftp.isi.edu/in-notes/rfc1952.txt>, Internet RFC 1952 (May 1996). The *zip* deflation format is specified in P. Deutsch, DEFLATE Compressed Data Format Specification version 1.3, <ftp://ftp.isi.edu/in-notes/rfc1951.txt>, Internet RFC 1951 (May 1996).

OPTIONS

-a --ascii

Ascii text mode: convert end-of-lines using local conventions. This option is supported only on some non-Unix systems. For MSDOS, CR LF is converted to LF when compressing, and LF is converted to CR LF when decompressing.

-c --stdout --to-stdout

Write output on standard output; keep original files unchanged. If there are several input files, the output consists of a sequence of independently compressed members. To obtain better compression, concatenate all input files before compressing them.

-d --decompress --uncompress

Decompress.

-f --force

Force compression or decompression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal. If the input data is not in a format recognized by *gzip*, and if the option *--stdout* is also given, copy the input data without change to the standard output: let *zcat* behave as *cat*. If *-f* is not given, and when not running in the background, *gzip* prompts to verify whether an existing file should be overwritten.

-h --help

Display a help screen and quit.

-l --list For each compressed file, list the following fields:

compressed size: size of the compressed file
uncompressed size: size of the uncompressed file
ratio: compression ratio (0.0% if unknown)
uncompressed_name: name of the uncompressed file

The uncompressed size is given as -1 for files not in gzip format, such as compressed .Z files. To get the uncompressed size for such a file, you can use:

```
zcat file.Z | wc -c
```

In combination with the *--verbose* option, the following fields are also displayed:

method: compression method
crc: the 32-bit CRC of the uncompressed data
date & time: time stamp for the uncompressed file

The compression methods currently supported are deflate, compress, lzh (SCO compress -H) and pack. The crc is given as ffffffff for a file not in gzip format.

With *--name*, the uncompressed name, date and time are those stored within the compress file if present.

With *--verbose*, the size totals and compression ratio for all files is also displayed, unless some sizes are unknown. With *--quiet*, the title and totals lines are not displayed.

-L --license

Display the *gzip* license and quit.

-n --no-name

When compressing, do not save the original file name and time stamp by default. (The original name is always saved if the name had to be truncated.) When decompressing, do not restore the original file name if present (remove only the *gzip* suffix from the compressed file name) and do not restore the original time stamp if present (copy it from the compressed file). This option is the default when decompressing.

-N --name

When compressing, always save the original file name and time stamp; this is the default. When decompressing, restore the original file name and time stamp if present. This option is useful on systems which have a limit on file name length or when the time stamp has been lost after a file transfer.

-q --quiet

Suppress all warnings.

-r --recursive

Travel the directory structure recursively. If any of the file names specified on the command line are directories, *gzip* will descend into the directory and compress all the files it finds there (or decompress them in the case of *gunzip*).

-S .suf --suffix .suf

Use suffix *.suf* instead of *.gz*. Any suffix can be given, but suffixes other than *.z* and *.gz* should be avoided to avoid confusion when files are transferred to other systems. A null suffix forces *gunzip* to try decompression on all given files regardless of suffix, as in:

```
gunzip -S "" *      (*. * for MSDOS)
```

Previous versions of *gzip* used the *.z* suffix. This was changed to avoid a conflict with *pack(1)*.

-t --test

Test. Check the compressed file integrity.

-v --verbose

Verbose. Display the name and percentage reduction for each file compressed or decompressed.

-V --version

Version. Display the version number and compilation options then quit.

-# --fast --best

Regulate the speed of compression using the specified digit #, where **-1** or **--fast** indicates the fastest compression method (less compression) and **-9** or **--best** indicates the slowest compression method (best compression). The default compression level is **-6** (that is, biased towards high compression at expense of speed).

ADVANCED USAGE

Multiple compressed files can be concatenated. In this case, *gunzip* will extract all members at once. For example:

```
gzip -c file1 > foo.gz
gzip -c file2 >> foo.gz
```

Then

```
gunzip -c foo
```

is equivalent to

```
cat file1 file2
```

In case of damage to one member of a .gz file, other members can still be recovered (if the damaged member is removed). However, you can get better compression by compressing all members at once:

```
cat file1 file2 | gzip > foo.gz
```

compresses better than

```
gzip -c file1 file2 > foo.gz
```

If you want to recompress concatenated files to get better compression, do:

```
gzip -cd old.gz | gzip > new.gz
```

If a compressed file consists of several members, the uncompressed size and CRC reported by the --list option applies to the last member only. If you need the uncompressed size for all members, you can use:

```
gzip -cd file.gz | wc -c
```

If you wish to create a single archive file with multiple members so that members can later be extracted independently, use an archiver such as tar or zip. GNU tar supports the -z option to invoke gzip transparently. gzip is designed as a complement to tar, not as a replacement.

ENVIRONMENT

The environment variable **GZIP** can hold a set of default options for *gzip*. These options are interpreted first and can be overwritten by explicit command line parameters. For example:

for sh: `GZIP="-8v --name"; export GZIP`

for csh: `setenv GZIP "-8v --name"`

for MSDOS: `set GZIP=-8v --name`

On Vax/VMS, the name of the environment variable is GZIP_OPT, to avoid a conflict with the symbol set for invocation of the program.

SEE ALSO

znew(1), zcmp(1), zmore(1), zforce(1), gzexe(1), zip(1), unzip(1), compress(1), pack(1), compact(1)

The *gzip* file format is specified in P. Deutsch, GZIP file format specification version 4.3, <<ftp://ftp.isi.edu/in-notes/rfc1952.txt>>, Internet RFC 1952 (May 1996). The *zip* deflation format is specified in P. Deutsch, DEFLATE Compressed Data Format Specification version 1.3, <<ftp://ftp.isi.edu/in-notes/rfc1951.txt>>, Internet RFC 1951 (May 1996).

DIAGNOSTICS

Exit status is normally 0; if an error occurs, exit status is 1. If a warning occurs, exit status is 2.

Usage: `gzip [-cdfhLnNrtvV19] [-S suffix] [file ...]`

Invalid options were specified on the command line.

file: not in gzip format

The file specified to *gunzip* has not been compressed.

file: Corrupt input. Use zcat to recover some data.

The compressed file has been damaged. The data up to the point of failure can be recovered using

```
zcat file > recover
```

file: compressed with *xx* bits, can only handle *yy* bits

File was compressed (using LZW) by a program that could deal with more *bits* than the decompress code on this machine. Recompress the file with gzip, which compresses better and uses less

memory.

file: already has .gz suffix -- no change

The file is assumed to be already compressed. Rename the file and try again.

file already exists; do you wish to overwrite (y or n)?

Respond "y" if you want the output file to be replaced; "n" if not.

gunzip: corrupt input

A SIGSEGV violation was detected which usually means that the input file has been corrupted.

xx.x% Percentage of the input saved by compression.

(Relevant only for **-v** and **-l**.)

-- not a regular file or directory: ignored

When the input file is not a regular file or directory, (e.g. a symbolic link, socket, FIFO, device file), it is left unaltered.

-- has *xx* other links: unchanged

The input file has links; it is left unchanged. See *ln(1)* for more information. Use the **-f** flag to force compression of multiply-linked files.

CAVEATS

When writing compressed data to a tape, it is generally necessary to pad the output with zeroes up to a block boundary. When the data is read and the whole block is passed to *gunzip* for decompression, *gunzip* detects that there is extra trailing garbage after the compressed data and emits a warning by default. You have to use the **--quiet** option to suppress the warning. This option can be set in the **GZIP** environment variable as in:

```
for sh: GZIP="-q" tar -xfz --block-compress /dev/rst0
```

```
for csh: (setenv GZIP -q; tar -xfz --block-compr /dev/rst0
```

In the above example, *gzip* is invoked implicitly by the **-z** option of GNU *tar*. Make sure that the same block size (**-b** option of *tar*) is used for reading and writing compressed data on tapes. (This example assumes you are using the GNU version of *tar*.)

BUGS

The *gzip* format represents the input size modulo 2^{32} , so the **--list** option reports incorrect uncompressed sizes and compression ratios for uncompressed files 4 GB and larger. To work around this problem, you can use the following command to discover a large uncompressed file's true size:

```
zcat file.gz | wc -c
```

The **--list** option reports sizes as **-1** and crc as **ffffff** if the compressed file is on a non seekable media.

In some rare cases, the **--best** option gives worse compression than the default compression level (**-6**). On some highly redundant files, *compress* compresses better than *gzip*.

COPYRIGHT NOTICE

Copyright © 1998, 1999, 2001, 2002 Free Software Foundation, Inc.

Copyright © 1992, 1993 Jean-loup Gailly

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

NAME

h2 – H" free Java SQL database engine

SYNOPSIS

edrc/bin/h2 [**-help** | **-?**]

h2 [**-url** *url*] [**-driver** *driver*] [**-user** *user*] [**-password** *password*] [**-web**] [**-tool**] [**-browser**] [**-tcp**] [**-pg**]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

H2 is a relational database management system written in Java. It can be embedded in Java applications or run in the client-server mode. The disk footprint (size of the jar file) is about 2.5 MB.

OPTIONS

-help usage message.

-? usage message.

-url *url* start a browser and connect to this URL

-driver *driver*
 used together with -url: the driver

-user *user* used together with -url: the user name

-password *password*
 used together with -url: the password

-web start the web server with the H2 Console

-tool start the icon or window that allows to start a browser

-browser start a browser connecting to the web server

-tcp start the TCP server

-pg start the PG server

For each server, additional options are available; for details, see the server tool.

ENVIRONMENT

See HTML H2 documentation in <http://www.h2database.com>.

EXIT STATUS

See HTML H2 documentation in <http://www.h2database.com>.

FILES

See HTML H2 documentation in <http://www.h2database.com>.

EXAMPLES

See HTML H2 documentation in <http://www.h2database.com>.

SEE ALSO

`edrcintro(1)`, <http://www.h2database.com>, <http://www.h2database.com/javadoc/org/h2/tools/Console.html>

NOTES

Parts of this manual page were extracted from the **H2** HTML documentation distributed with the H2 version 1.3.172 and the Wikipedia page [http://en.wikipedia.org/wiki/H2_\(DBMS\)](http://en.wikipedia.org/wiki/H2_(DBMS)) written by "The H2 Development Team" led by Thomas Mueller.

BUGS

-

AUTHOR

h2 was developed by Thomas Mueller and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

histlist – print shell history files

SYNOPSIS

edrc/lib/histlist [**-h**]

histlist [*histfile* [*number*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print a table of history files including the resolution of usage begin, usage end, history file size and history file name.

The begin can be resolved properly only if the history file names comply to the history file names as produced by the **shell**(1) command when started outside **edrc**.

OPTIONS

-h usage message.

histfile history file. The **histlist** command searches the directory of the given *histfile* for other history files that will be listed by the command.

number number of most recent history files listed.

ENVIRONMENT**\$HISTFILE**

history file.

EXIT STATUS

0 no error

4 usage listed.

FILES

`~<USER>/eshell/sh_history/./<MIL-DATE>_<TIME>.<TTY>`
default history file.

EXAMPLES

-

SEE ALSO

edrcintro(1), **eshell(1)**, **shell(1)**

NOTES

when calling the **histlist** command within **shell(1)** or **eshell(1)** you are actually calling an alias having the definition:

```
alias histlist='\histlist $HISTFILE 100'
```

BUGS

-

AUTHOR

histlist was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

homedir – Get Home Directory of a User

SYNOPSIS

edrc/lib/homedir [*user*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **homedir** command returns the logical path of a given user to standard output. In the C-shell and the Korn-shell this corresponds to **cd** *~user* . The **homedir user** command is especially useful in Bourne-shell scripts, where the above command with *~* does not work.

OPTIONS

user username of the user whose homedir has to be resolved.

EXIT STATUS

0 always

EXAMPLES

1) HOWTO use homedir to evaluate a home directory of a user

```
[ /home/user ]  
[ user@client ] [ksh]: homedir sfitools  
/apl/sfitools
```

SEE ALSO

edrcintro(1), **gid(3)**, **gecos(3)**, **group(3)**, **uid(3)**, **user(3)**

FILES

/etc/passwd
(or related NIS/NIS+ table)

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

BUGS

-

AUTHOR

homedir was developed by Peter Stevens and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

hostaliases – return aliases for a hostname

SYNOPSIS

edrc/lib/hostaliases [**-h**]

hostaliases *hostname*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return all defined aliases for a given hostname.

OPTIONS

-h print usage message.

hostname name of a host.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 no aliases found for a given host.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **gethostbyname(3)**, **host(1)**, **nslookup(1)**

NOTES

-

BUGS

IPv6 is not supported yet.

AUTHOR

hostaliases was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

hostlist – return a list of hosts

SYNOPSIS

edrc/lib/hostlist [**-h**]

hostlist

hostlist [**-g** *hostgroupname*{[,*hostgroupname*]}][**-l**]

hostlist -p

hostlist -i | -y | -j

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **hostlist** command returns a list of hosts.

The intention is, that the output generated by **hostlist** is used as input for scripts or commands which compute commands/jobs on this list of hosts. Therefore there is only one source list of hosts being part of an environment and when a host is added/removed to/from it, only the **edrc/etc/hostlist.cfg** configuration file has to be adjusted and not all scripts which have to deal with a list of hosts.

hostlist can be used together with the **server_environment**(3) command in the configuration file **edrc/etc/hostlist.cfg** to enable **hostlist** to return the list of hosts that are part of a certain named server environment. E.g. the list of hosts in the PRODUCTION environment is different from the list of hosts in the DEVELOPMENT environment for an enterprise. See **hostlist.cfg**(4) for more information.

To compute different sets of hosts as returned by a call of **hostlist**, it is possible to define groups of hosts that can be accessed via a *hostgroupname*. This enables the abstraction of a group of hosts to a group name (*@HOSTGROUP*).

On complex configurations the resolution performance might decrease. To avoid this, the resolution cache can be used (**CACHE=True** in **edrc/etc/hostlist.cfg**). The cache data depends on the timestamps and file-sizes of a configurable set of files (**CACHE_DEPENDENCY=" file_1 file_2 ... file_n-1 file_n "**), the **hostlist**(3) and **server_environment**(3) command files, and the hostname as resolved by **hostname**(1). The resolution cache is read/write for the WA2L/edrc owner and read only for all other users.

OPTIONS

return the list of hosts being part of the current server environment. See also **server_environment**(3) for information about the definition of server environments.

- h** usage message.

- g** *hostgroupname_list*
 comma separated list of *hostgroupname*. The convention is, that *hostgroups* always start with the @ character.

- l** the returned list of hosts is comma separated instead of space separated (=default). This eases further processing of the output as handing over the output to the **-t** option of **rcmd**.

- p** print all configured host groups in human readable form. The item listed without a *HOSTGROUP* name is the output of **hostlist** when used without an option.

 The *HOSTGROUPS* having a name starting with an underscore (_) should **not** be used interactively, due to the fact that the convention is, that those *HOSTGROUPS* are used internally within configuration files only.

- i** print all configured host groups as Ansible inventory ini file format.

 The *HOSTGROUPS* having a name starting with an underscore (_) are *HOSTGROUPS* that are used internally in WA2L/edrc within configuration files.

- y** print all configured host groups as Ansible inventory yaml file format.

 The *HOSTGROUPS* having a name starting with an underscore (_) are *HOSTGROUPS* that are used internally in WA2L/edrc within configuration files.

- j** print all configured host groups as Ansible inventory json file format.

 The *HOSTGROUPS* having a name starting with an underscore (_) are *HOSTGROUPS* that are used internally in WA2L/edrc within configuration files.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.

- 2** a file that is defined in the **CACHE_DEPENDENCY** file list does not exist.

- 4** usage printed.

- 6** configuration file **edrc/etc/hostlist.cfg** does not exist.

FILES

edrc/etc/hostlist.cfg

configuration file of **hostlist**. this file contains all host list and host group definitions.

edrc/var/cache/hostlist/

resolution cache of the **hostlist** command if the **CACHE** setting in the configuration file **edrc/etc/hostlist.cfg** is set to *True*.

edrc/var/cache/hostlist/meta

meta data for the resolution cache. This file is used to check if the cache data is still valid.

edrc/var/cache/hostlist/data

this file contains the cache data, the resolved **HOSTLIST** and **HOSTGRPS** settings for the system where **hostlist** has been executed.

edrc/var/samples/hostlist/

configuration examples for **hostlist(3)** and **server_environment(3)**.

EXAMPLES

1) command line usage example

```
[ /home/wilma ]
[ wilma@adm_db1_prod ][-ksh]: ~edrc/lib/hostlist

dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi32 dcdbsi33
```

2) command line usage example using a hostgroup

```
[ /home/wilma ]
[ wilma@adm_db1_prod ][-ksh]: ~edrc/lib/hostlist -g @ADM

adm_db1_mnt adm_db1_tst adm_db1_pre adm_db1_prod
```

3) use of hostlist in configuration files

The use of **hostlist** in configuration files eases the configuration due to the fact that the target list of hosts can be centralized to one single configuration file (the **edrc/etc/hostlist.cfg** file) and changes in the server environment will be propagated to all tools used.

Example of the use of **hostlist** in the **edrc/etc/edrc.cfg** configuration file:

```
:
:

# Space separated list of hosts. To this hosts
# the files in SCRIPTS_BASEDIR will be distributed.
#
DIST_HOSTLIST="`hostlist -g @_DIST`"

:
```

:

Example of the use of **hostlist** in the **edrc/etc/passwdsyncd.cfg** configuration file:

:

:

```
# Space separated list of hosts. To this hosts
# the password information will be distributed.
#
SYNC_HOSTLIST="'hostlist'"
```

:

:

4) usage in scripts

Print uptime of all systems in a server environment.

:

:

```
targets='hostlist -l'
rcmd -n -t $targets -c "sysinfo -u" | sort
```

:

:

SEE ALSO

edrcintro(1), **hostlist.cfg**(4), **hostlistdat2cfg**(3), **rcmd**(1), **resolve_targetlist**(3), **server_environment**(3), **edrc/var/samples/hostlist/**

NOTES

The commands **rcmd**(3) and **filelist**(1) use **hostlist** internally to resolve the list of hosts being part of the server environment where logged in.

BUGS

-

AUTHOR

hostlist was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

hostlist.cfg – configuration file for hostlist

SYNOPSIS

edrc/etc/hostlist.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **hostlist** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**HOSTLIST**

Definition of hosts that are returned when invoking **hostlist** without options.

Example: HOSTLIST="dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi32 dcdbsi33"

Default: HOSTLIST=""

HOSTGRPS

Definitions of groups of hosts that are returned when invoking **hostlist** with the **-g @GRP-NAME** option.

Example: HOSTGRPS="@ADM: adm_db1_mnt adm_db1_tst adm_db1_pre adm_db1_prod
@BE: dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi32"

Default: HOSTGRPS=""

CACHE Set this setting to *True* to cache the resolved **HOSTLIST** and **HOSTGRPS** settings, set it to *False* to not to use a cache.

Example: CACHE=True

Default: CACHE=False

CACHE_DEPENDENCY

List of files on which the cache data depends on. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation. This setting is only active if the **CACHE** setting is set to *True*.

Example: CACHE_DEPENDENCY="etc/hostlist.cfg etc/hostlist.dat etc/server_environment.cfg"

Default: CACHE_DEPENDENCY="etc/hostlist.cfg etc/server_environment.cfg"

EXAMPLES

1) very simple configuration for one environment

This configuration contains the servers in the environment (PRODUCTION) of an enterprise where WA2L/edrc is installed. If WA2L/edrc is installed in other environments (TEST, PRE-PRODUCTION), too, a specific **edrc/etc/hostlist.cfg** configuration file has to be defined for each environment.

```
#
# Purpose of the different HOSTGRPS:
#
#   @_DIST:      Distribution group where edrc distributes when invoking t
#                 edrc 'distribute' command. Do not use this group for manu
#                 invocations of rcmd or filedist.

# List of hosts / Groups of hosts
#
HOSTLIST="dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi32 dcdbsi33"
HOSTGRPS="@_DIST:
                                dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi3
                                dcdbsi33
"
```

2) advanced config for multiple environments

To have a single **edrc/etc/hostlist.cfg** configuration file for all environments (TEST, PREPRODUCTION, PRODUCTION) in the enterprise the **server_environment(3)** command can be used in the configuration file. In this configuration file example the **HOSTLIST** definition is defined for each environment with the focus to the related environment and because the **HOSTGRPS** definition is identical for all environments it can be defined once. If **HOSTGRPS** are different for each environment it has to be defined for each environment, too.

```
#
# Purpose of the different HOSTGRPS:
#
```

```

#   @_DIST:      Distribution group where edrc distributes when invoking t
#                 edrc 'distribute' command. Do not use this group for manu
#                 invocations of rcmd or filedist.
#   @ALL:        All servers of the whole environment (MAINTENANCE, TEST,
#                 PREPRODUCTION and PRODUCTION)
#
# Groups of hosts
#
HOSTGRPS="@_DIST:
                                dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi3
                                dcdbsi33
                                dcdbsi61 dcdbsi62 dcdbsi63 dcdbsi64
                                dcdbsi71 dcdbsi72 dcdbsi73 dcdbsi74
                                dcdbsi50
                                @ALL:
                                dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi3
                                dcdbsi33
                                dcdbsi61 dcdbsi62 dcdbsi63 dcdbsi64
                                dcdbsi71 dcdbsi72 dcdbsi73 dcdbsi74
                                "

# List of hosts
#
case `server_environment` in
    PRODUCTION)
        HOSTLIST="dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi32
        ;;
    PREPRODUCTION)
        HOSTLIST="dcdbsi61 dcdbsi62 dcdbsi71 dcdbsi72"
        ;;
    TEST)
        HOSTLIST="dcdbsi63 dcdbsi64 dcdbsi73 dcdbsi74"
        ;;
esac

```

3) advanced config for multiple environments with disaster recovery support

The setup in this disaster recovery scenario is, that the PREPRODUCTION is used to recover the PRODUCTION environment to.

If the file `./DISASTER_CASE` exists it is assumed that a DISASTER CASE SITUATION exists. In that case the list of hosts is returned that correlated to this situation. To make this happen the configuration `edrc/etc/server_environment.cfg` has to be replaced by a configuration file that returns PRODUCTION for the hosts being part of the former PREPRODUCTION environment.

```

# List of hosts / Groups of hosts
#
case `server_environment` in
    PRODUCTION)
        if [ -f ./DISASTER_CASE ]; then
            ##

```

```

## DISASTER CASE SITUATION
## All servers from the PREPRODUCTION are used to for
## PRODUCTION environment.
## The HOSTGRPS are restricted PREPRODUCTION host
## encapsulate the 'Disaster Recovery Environment'
## the other environments during a 'Disaster Simul
## or a 'Disaster Case'.
##
##
HOSTLIST="dcdbsi61 dcdbsi62 dcdbsi71 dcdbsi72"
HOSTGRPS="@_DIST:
                                dcdbsi61 dcdbsi62 dcdbsi7
                                dcdbsi50
                                @ALL:
                                dcdbsi61 dcdbsi62 dcdbsi7
                                "
else
##
## NORMAL OPERATION SITUATION
##
HOSTLIST="dcdbsi20 dcdbsi21 dcdbsi30 dcdbsi31 dcdbsi32
HOSTGRPS="@_DIST:
                                dcdbsi20 dcdbsi21 dcdbsi3
                                dcdbsi33
                                dcdbsi61 dcdbsi62 dcdbsi6
                                dcdbsi71 dcdbsi72 dcdbsi7
                                dcdbsi50
                                @ALL:
                                dcdbsi20 dcdbsi21 dcdbsi3
                                dcdbsi33
                                dcdbsi61 dcdbsi62 dcdbsi6
                                dcdbsi71 dcdbsi72 dcdbsi7
                                "
fi
;;

:
:

esac

```

4) define hosts in hostlist.dat

This is the definition as documented in **edrcsetup**(1m) of WA2L/edrc starting with release 1.5.43 and is also the configuration file contents that is distributed with the full application package from now on (see also: **edrc/var/samples/hostlist/07/** and **edrc/var/samples/hostlist/08/**).

To define the hosts and groups in the **hostlist.dat** file, set the **USE_HOSTLIST_DAT** setting to *True*.

```

:
:

```

```

# Set USE_HOSTLIST_DAT to True, if hosts are listed in
# the etc/hostlist.dat file, if hosts are defined directly
# in etc/hostlist.cfg, set USE_HOSTLIST_DAT to False.
#
USE_HOSTLIST_DAT=True

# Define the central administration environment in the
# format CUSTOMER.NAME as defined in server_environment.cfg.
#
CMAN_ENVIRONMENT="Highlander.ADMIN"

# Read hostlist.dat definitions as resolved.
#
if [ "$USE_HOSTLIST_DAT" = True ]; then
    case `server_environment -C`.`server_environment` in
        $CMAN_ENVIRONMENT)
            out=`hostlistdat2cfg -o -m MGRPS_LIST=@LINUX,@SOLARIS,@
            *) out=`hostlistdat2cfg -o` ;;
        esac
        . $out; removetemp -q -d `dirname ${out}`; return 0
    fi

# List of hosts when defined directly in the hostlist.cfg
# file.
#
case `server_environment -C`.`server_environment` in
    WA2L.EDRC_DVLP)
        HOSTLIST="
        "
        HOSTGRPS="@_DIST:
        @ALL:
        "
        ;;
    esac

```

SEE ALSO

edrcintro(1), hostlist(1), hostlist.dat(4), hostlistdat2cfg(3), server_environment(3), server_environment.cfg(4), edrc/var/samples/hostlist/

NOTES

-

BUGS

-

AUTHOR

hostlist.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

hostlist.dat – list of hosts for the hostlist command

SYNOPSIS

edrc/etc/hostlist.dat

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the host list definition for the **hostlistdat2cfg** command which converts this data file to a format that can be computed by the **hostlist** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

CUSTOMER;ENVIRONMENT;GROUPS;OPTIONS;HOSTS;

Rows starting with a # are considered as comments. Empty lines are allowed, too.

OPTIONS

CUSTOMER

customer name as defined in **edrc/etc/server_environment.cfg**.

ENVIRONMENT

customer environment name as defined in **edrc/etc/server_environment.cfg**.

GROUPS

space separated list of host groups.

OPTIONS

Additional options for a row. Currently this field is not resolved and should be left empty.

HOSTS

space separated list of hosts.

EXAMPLES

```
#
# etc/hostlist.dat - hostlist csv database file
#
# [00] 12.08.2008 CWa    Initial Version
#
#
# Format:
#
# <CUSTOMER>;<ENVIRONMENT>;<GROUPS>;<OPTIONS>;<HOSTS>;
#
# To verify the syntax, use: hostlistdat2cfg -a list_dat
# Only correct entries will be listed.
#
Highlander ;ADMIN          ;          ;;adm-linux adm-hpux adm-solaris;
Highlander ;REPORT        ;          ;;rpt-001 rpt-002;
ACME       ;DEVELOPMENT   ;@APP      ;;host-001 host-002;
ACME       ;DEVELOPMENT   ;@DB       ;;host-003;
ACME       ;TEST          ;          ;;host-101 host-103;
ACME       ;PREPRODUCTION ;          ;;host-201 host-203 host-205;
ACME       ;PRODUCTION    ;          ;;host-303 host-308 host-309;
ACME       ;              ;@PING     ;;ping-001 ping-002 ping-003;
#
WA2L       ;EDRC_DVLP      ;@DEV      ;;tobago \
                ub2004 \
                suse110 suse110-32 suse113 \
                suse114 suse123 suse152;
```

SEE ALSO

edrcintro(1), **hostlist(3)**, **hostlist.cfg(4)**, **hostlistdat2cfg(3)**, **server_environment(3)**, **server_environment.cfg(4)**

NOTES

the reason why the **hostlist.dat** file is converted using **hostlistdat2cfg** into a file that can be computed by the **hostlist** command is, that first the **hostlist.cfg** existed and over time it turned out to be too complicated to maintain, but a backward compatible solution was needed.

Therefore it is recommended to define the list of hosts in the **hostlist.dat** file and not any more in the **hostlist.cfg** file.

BUGS

-

AUTHOR

hostlist.dat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

hostlistdat2cfg – generate hostlist.cfg from hostlist.dat

SYNOPSIS

edrc/lib/hostlistdat2cfg [**-h**]

hostlistdat2cfg [**-o**]

hostlistdat2cfg [**-f** *file* ...] [*central_opt*] [*action_opt*] [*env_opt*]

central_opt ::= **-m** [[**XGRPS_PREF**=*string*] [**XGRPS_LEGACY**] [**MGRPS_LIST**=*hostgroup*list]]

action_opt ::= **-a** { [**print_cfg**] [**-o**] | **list_dat** [*type_opt*] | **list_res** [*type_opt*] }

env_opt ::= **-n** *CUSTOMER.NAME*

type_opt ::= **-t** { **table** | **csv** | **csv_header** }

AVAILABILITY

WA2L/edrc

DESCRIPTION

convert a **hostlist.dat** file that has the structure

Example:

```
#
# Format:
#
#   <CUSTOMER>; <ENVIRONMENT>; <GROUPS>; <OPTIONS>; <HOSTS>
#
ACME;DEVELOPMENT      ;@APP      ;;host-001 host-002;
ACME;DEVELOPMENT      ;@DB       ;;host-003;
ACME;TEST              ;@APP      ;;host-101 host-103;
ACME;PREPRODUCTION    ;@APP      ;;host-201 host-202 host-205;
ACME;PRODUCTION        ;@APP      ;;host-303 host-308 host-309;
```

into settings that can be used in the **hostlist.cfg** file.

This command will significantly simplify the definition of host lists of large environments and will make the more complicated constructs obsolete where using the **fmatch**(3) command in the **hostlist.cfg**(4) configuration file.

Despite the many options that the **hostlistdat2cfg** provides, normally only the **-m** and the **-o** options will be used in the **hostlist.cfg** config file. See also samples in the **edrc/var/samples/hostlist** directory.

OPTIONS

-h print usage message.

-o write configuration to temporary file and return the name of the the temporary file.

The output is written to a temporary directory that is generated by the **maketemp(3)** command internally. Therefore the tempfile can be removed using **removetemp(3)** later:

```
out=`hostlistdat2cfg -o`
. $out
removetemp -q -d `dirname ${out}`
```

-f file... a list of csv files. If this option is not specified, the **edrc/etc/hostlist.dat** file is read by default.

The csv files is a semicolon separated file having the format as explained below. Rows can span multiple lines in the file, if they end with a backslash (\).

To verify the syntax of the csv file, use the command **hostlistdat2cfg -a list_dat -f file** which will only list the correct rows.

Format:

CUSTOMER;ENVIRONMENT;GROUPS;OPTIONS;HOSTS;

Keys:

CUSTOMER

Customer name as returned by **server_environment -C**.

ENVIRONMENT

Environment name as returned by **server_environment**. If this field is left empty, the row is valid for all environments of the *CUSTOMER*.

GROUPS Space separated list of hostgroups (*@HOSTGRP*) as known from the **hostlist(3)** command.

The *GROUPS* *@ALL* and *@_DIST* are generated automatically out of all hosts specified in the *HOSTS* field for the *ENVIRONMENT*s of a *CUSTOMER*.

If certain host(s) should be excluded from the *@ALL* or *@_DIST* group, specify *-@ALL* or *-@_DIST* in the *GROUPS* field.

OPTIONS Additional options for a row. Currently this field is not resolved and should be left empty.

HOSTS space separated list of hosts.

-m [*options*]

options:

XGRPS_PREF=*string*

cross env HOSTGRPS prefix (default="")

XGRPS_LEGACY

also create legacy customer HOSTGRPS omitting **.ALL**.

MGRPS_LIST=*hostgroup**list*

comma separated HOSTGRPS to be merged.

-a *action* action:

print_cfg print the **hostlist.cfg** configuration structure. If no action is given, this is the default action.

list_dat list the data of the given csv (**hostlist.dat**) file(s). This option can be used to check if the syntax of the processed csv file(s) is correct, because only correct columns are printed.

list_res list resolved **hostlist.dat** file for the related environment. This option can also be used for further data processing.

-t *type* type:

table list in table format. If no type is given, this is the default output type.

csv list in csv format. When using this option, only the data rows are listed.

csv_header

list the header row for the **csv** output.

-n *CUSTOMER.NAME*

set the *CUSTOMER* and *ENVIRONMENT* name.

This enables to test a configuration for an certain environment without logging in to it.

If this option is not used, the default is the customer name and environment of the server where logged on as returned by **server_environment -C** (=CUSTOMER) and **server_environment** (=NAME).

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|------------------------|
| 0 | no error. |
| 4 | usage message printed. |

FILES**edrc/etc/hostlist.dat**configuration file of **hostlistdat2cfg**. This file contains all host list and host group definitions.**EXAMPLES****1) print config for environment where logged on:**`hostlistdat2cfg`**2) print central management config for environment where logged on:**`hostlistdat2cfg -m`**3) print central management config for environment where logged on and merge all operating system groups (@SOLARIS, @LINUX ...) that are defined in all customer environments:**`hostlistdat2cfg -m MGRPS_LIST=@LINUX,@SOLARIS,@HPUX`

With that the local @ANY.SOLARIS group will contain all hosts that are member of @SOLARIS groups in all customer environments.

4) use hostlistdat2cfg in etc/hostlist.dat:

```
USE_HOSTLIST_DAT=True
CMAN_ENVIRONMENT="Highlander.ADMIN"

if [ "$USE_HOSTLIST_DAT" = True ]; then
  case `server_environment -C`.`server_environment` in
    $CMAN_ENVIRONMENT)
      out=`hostlistdat2cfg -o -m MGRPS_LIST=@LINUX,@SOLARIS,@HPUX`;
      *) out=`hostlistdat2cfg -o`;
    esac
  . $out; removetemp -q -d `dirname ${out}`; return 0
fi
```


5) example with complete file set server_environment.cfg, hostlist.cfg and hostlist.dat

edrc/etc/server_environment.cfg

```
#
# server_environment.cfg - config for server_environment
#
# [00] 24.07.2004 CWa Initial Version
# [50] 18.04.2013 CWa chg: cleanup
#

# Server environment definition.
#
# If you log on to a system, the <NAME> field of the
# first matching <server_regex> will be returned.
#

#
# Format:
#   <NAME>:<Description>:<server_regex>:<Customer>:
#

# Highlander (the 'one and only' central administration environment)
#
ADMIN:Datactr Schaffhausen, Central ADMIN:adm-(linux|hpux|solaris):High
REPORT:Datactr Schaffhausen, Central ADMIN:rpt-[0-9]+):Highlander:

# ACME
#
PRODUCTION:Datacenter Bern, PRODUCTION Env:host-3[0-9][0-9]:ACME:
PREPRODUCTION:Datacenter Boston, Integration Env:host-2[0-9][0-9]:ACME:
TEST:Datacenter Plano, Test Env:host-1[0-9][0-9]:ACME:
DEVELOPMENT:Datacenter San Jose, Development Env:host-0[0-9][0-9]:ACME:
```

edrc/etc/hostlist.cfg

```
#
# etc/hostlist.cfg - configuration of hostlist
#
# [00] 16.02.2004 CWa Initial Version
# [64] 16.03.2013 CWa chg: use of hostlistdat2cfg
#

# Set USE_HOSTLIST_DAT to True, if hosts are listed in
# the etc/hostlist.dat file, if hosts are defined directly
# in etc/hostlist.cfg, set USE_HOSTLIST_DAT to False.
#
USE_HOSTLIST_DAT=True

# Define the central administration environment in the
# format CUSTOMER.NAME as defined in server_environment.cfg.
#
```

```

CMAN_ENVIRONMENT="Highlander.ADMIN"

# Read hostlist.dat definitions as resolved.
#
if [ "$USE_HOSTLIST_DAT" = True ]; then
    case `server_environment -C`.`server_environment` in
        $CMAN_ENVIRONMENT)
            out=`hostlistdat2cfg -o -m MGRPS_LIST=@LINUX,@SOLARIS,@HPUX`
            *) out=`hostlistdat2cfg -o`;
        esac
        . $out; removetemp -q -d `dirname ${out}`; return 0
    fi

# List of hosts when defined directly in the hostlist.cfg
# file.
#
# The only compulsory settings in the hostlist.cfg file are
# HOSTLIST and HOSTGRPS, everything else is for convenience
# only and could be removed. See also: hostlist.cfg (4).
#
case `server_environment -C`.`server_environment` in
    Dummy.DUMMY)
        HOSTLIST="
"
        HOSTGRPS="@_DIST:
@ALL:
"
        ;;
    esac

```

edrc/etc/hostlist.dat

```

#
# etc/hostlist.dat - hostlist csv database file
#
# [00] 12.08.2008 CWa Initial Version
# [58] 16.03.2013 CWa chg: to hostlistdat2cfg structure
#
#
# Format:
#
# <CUSTOMER>;<ENVIRONMENT>;<GROUPS>;<OPTIONS>;<HOSTS>
#
# To verify the syntax, use: hostlistdat2cfg -a dat
# Only correct entries will be listed.
#
Highlander ;ADMIN ; ;;adm-linux adm-hpux adm-solaris;
Highlander ;REPORT ; ;;rpt-001 rpt-002;
ACME ;DEVELOPMENT ;@APP ;;host-001 host-002;
ACME ;DEVELOPMENT ;@DB ;;host-003;
ACME ;TEST ; ;;host-101 host-103;

```

```
ACME      ;PREPRODUCTION ;      ;;host-201 host-203 host-205;
ACME      ;PRODUCTION   ;      ;;host-303 host-308 host-309;
ACME      ;              ;@PING ;;ping-001 ping-002 ping-003;
```

resulting configuration on ACME.TEST

```
#
# hostlist.cfg - configuration for hostlist
#
# [00] 03.05.2013 root generated by: /opt/edrc/lib/hostlistdat2cfg
#
# resolved for:
#
# SERVER_ENVIRONMENT_CUSTOMER: ACME
# SERVER_ENVIRONMENT_NAME: TEST
#
CACHE=True

CACHE_DEPENDENCY="etc/hostlist.cfg etc/server_environment.cfg
                  etc/hostlist.dat lib/hostlistdat2cfg"

HOSTLIST="
          host-101 host-103
"

HOSTGRPS="
@_DIST:
          host-001 host-002 host-003
          host-101 host-103
          host-201 host-203 host-205
          host-303 host-308 host-309

@PING:
          ping-001 ping-002 ping-003

@ALL:
          host-001 host-002 host-003
          host-101 host-103
          host-201 host-203 host-205
          host-303 host-308 host-309
          ping-001 ping-002 ping-003
"

export HOSTLIST HOSTGRPS CACHE CACHE_DEPENDENCY
```

resulting configuration on Highlander.ADMIN

```
#
# hostlist.cfg - configuration for hostlist
#
# [00] 03.05.2013 root generated by: /opt/edrc/lib/hostlistdat2cfg
#
# resolved for:
#
```

```

# SERVER_ENVIRONMENT_CUSTOMER: Highlander
# SERVER_ENVIRONMENT_NAME:      ADMIN
#
CACHE=True

CACHE_DEPENDENCY="etc/hostlist.cfg etc/server_environment.cfg
                  etc/hostlist.dat lib/hostlistdat2cfg"

HOSTLIST="
          adm-hpux adm-linux adm-solaris
"

HOSTGRPS="
@Highlander.ADMIN:
          adm-hpux adm-linux adm-solaris

@ACME.TEST:
          host-101 host-103

@ACME.PREPRODUCTION:
          host-201 host-203 host-205

@ACME.PRODUCTION:
          host-303 host-308 host-309

@ACME.DEVELOPMENT:
          host-001 host-002 host-003

@ACME.ALL:
          host-001 host-002 host-003
          host-101 host-103 host-201
          host-203 host-205
          host-303 host-308 host-309
          ping-001 ping-002 ping-003

@ANY.ALL:
          adm-hpux adm-linux adm-solaris
          host-001 host-002 host-003
          host-101 host-103
          host-201 host-203 host-205
          host-303 host-308 host-309
          ping-001 ping-002 ping-003
          rpt-001 rpt-002

@ACME.DEVELOPMENT.APP:
          host-001 host-002

@Highlander.ALL:
          adm-hpux adm-linux adm-solaris
          rpt-001 rpt-002

@_DIST:
          adm-hpux adm-linux adm-solaris
          rpt-001 rpt-002

```

```

@Highlander.REPORT:
    rpt-001 rpt-002

@ACME.DEVELOPMENT.DB:
    host-003

@ACME.ANY.PING:
    ping-001 ping-002 ping-003

@ALL:
    adm-hpux adm-linux adm-solaris
    rpt-001 rpt-002

"
export HOSTLIST HOSTGRPS CACHE CACHE_DEPENDENCY

```

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **hostlist(3)**, **hostlist.cfg(4)**, **hostlist.dat(4)**, **server_environment(3)**, **edrc/var/samples/hostlist/**

NOTES

-

BUGS

-

AUTHOR

hostlistdat2cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

html2mht – convert a HTML file to a single Web Archive file (MHT)

SYNOPSIS

edrc/lib/html2mht [**-h**]

html2mht **-i** *htmlfile* **-o** *mhtfile* [**-f**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

convert a HTML file that has referenced images to a single web archive file (mht) that contains the image files.

The MHT file can be viewed using the Microsoft Internet Explorer (TM) or an other MHT viewer.

OPTIONS

-h print usage message.

-i *htmlfile* input HTML file to be converted.

-o *mhtfile* output MHT file. To be able to read the file directly by the Microsoft Internet Explorer (TM) the *mhtfile* should be named *filename.mht* .

-f force overwriting of the output *mhtfile*.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 *mhtfile* does already exist.

2 *htmlfile* cannot be read.

- 4 usage message printed.
- 5 cannot write to output file *mhtfile*.
- 6 cannot write to output directory of *mhtfile*.

FILES

edrc/var/samples/html2mht/
example files to test **html2mht(3)**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc/var/samples/html2mht/**

NOTES

-

BUGS

CSS files and JavaScript files have to be included in the *htmlfile* to be converted.

AUTHOR

html2mht was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME**SYNOPSIS**

edrc/bin/hwinventory [**-h**]

hwinventory [**-d**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

command to print a minimal set of inventory information needed for asset management.

OPTIONS

-h usage message.

-d **CSV output.**

-t print **CSV** header.

ENVIRONMENT

-

EXIT STATUS

0 always.

usage message displayed.

FILES

-

EXAMPLES

Example usage of the **hwinventory** command:

```
[ / ]  
[ root@acme-007 ][*edrc*/bash]: hwinventory
```



```

SERVERNAME ..... : acme-007
DNSDOMAIN ..... : home
CUSTOMER ..... : WA2L
ENVIRONMENT ..... : EDRC_DVLP
DESCRIPTION ..... : Beringen, EDRC Development
OSID ..... : Linux-64
OSNAME ..... : Linux
OSVERSION ..... : 5.4.0-52-generic
OSDISTRIBUTION .... : Ubuntu
ARCHITECTURE ..... : x86_64
MODEL ..... : Latitude E4300
SERIALNUMBER ..... : JYHKS4J
HOSTID ..... : 007a0102
CPU ..... : 2
MEMORY ..... : 3916368 kB
IPADDRESS ..... : 5a01:1206:e6b3:2090:2419:8fa6:0a6:8684
LOCALDISK ..... : ?
EDRC ..... : 1.5.54
TIMEDAT ..... : 2020-11-09 18:30:30
TIMESTAMP ..... : 1604943030

```

SEE ALSO

edrcintro(1), **cfg2html**(1m), **swvi**(1)

NOTES

For a comprehensive configuration documentation, use the **cfg2html**(1m) command.

BUGS

-

AUTHOR

hwinventory was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

indent – continuous unbuffered output with an indent

SYNOPSIS

edrc/lib/indent [*width* [*output_channel*]]

(deprecated: **ident** [*width*])

AVAILABILITY

WA2L/edrc

DESCRIPTION

continuous output of characters received via **stdin** with a leading indent (*width*) on each line.

OPTIONS

width width of output indent. If not specified the default of 4 applies.

output_channel

if set to **1** the leading indent goes to **stdout** which is the default. If set to **2** the leading indent goes to **stderr**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

FILES

-

EXAMPLES**1) indented output of a file**

Print a file with the standard indent of 4:

```
[ / ]
[ root@acme007 ][-sh]: cat /etc/group | indent
      root:x:0:root,HMUST
```

```
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
:
:
```

Print the file with a specific indent of 8:

```
[ / ]
[ root@acme007 ][-sh]: cat /etc/group | indent 8
    root:x:0:root,HMUST
    bin:x:1:root,bin,daemon
    daemon:x:2:root,bin,daemon
    sys:x:3:root,bin,adm
    :
    :
```

2) indented output in a script

In the following script produces an output as:

```
:
:
interface files ...
    csv files in '/data/incoming' ...
        /data/incoming/HR/12002.csv
        /data/incoming/HR/12003.csv
        /data/incoming/HR/12004.csv
        /data/incoming/HR/12005.csv
        /data/incoming/GL/50003.csv
        /data/incoming/GL/50004.csv
    done.
    csv files in '/data/outgoing' ...
        /data/outgoing/HR/12001.csv
        /data/outgoing/GL/50001.csv
        /data/outgoing/GL/50002.csv
    done.
done.
:
:
```

Be aware of the 'cascaded' usage of the **indent** command first to indent the output of the **find** command and then to indent the output of the whole **search_csv()** function.

To ensure, that error output of the **find** command output is also indented, the output of **find** to **stderr** has to be redirected to **stdout** using the **2>&1** redirection syntax.

```
#!/bin/sh
:
:
search_csv(){ start=$1
    echo "csv files in '$start' ..."
}
```

```

        find $start -name *.csv -print 2>&1 | indent
        echo "done."
    } # search_csv

main() {

    in=/data/incoming
    out=/data/outgoing
    :
    :
    echo "interface files ..."
    search_csv $in | indent
    search_csv $out | indent
    echo "done."
    :
    :
} # main
main $*
```

The **main()** function above could also be written as:

```

main() {

    in=/data/incoming
    out=/data/outgoing
    :
    :
    echo "interface files ..."
    {
        search_csv $in
        search_csv $out
    } | indent
    echo "done."
    :
    :
} # main
```

SEE ALSO

edrcintro(1), **cat(1)**, **csv(3)**, **sh(1)**

NOTES

-

BUGS

-

AUTHOR

indent was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

indent – continuous unbuffered output with an indent

SYNOPSIS

edrc/lib/indent [*width* [*output_channel*]]

(deprecated: **ident** [*width*])

AVAILABILITY

WA2L/edrc

DESCRIPTION

continuous output of characters received via **stdin** with a leading indent (*width*) on each line.

OPTIONS

width width of output indent. If not specified the default of 4 applies.

output_channel

if set to **1** the leading indent goes to **stdout** which is the default. If set to **2** the leading indent goes to **stderr**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

FILES

-

EXAMPLES**1) indented output of a file**

Print a file with the standard indent of 4:

```
[ / ]
[ root@acme007 ][-sh]: cat /etc/group | indent
      root:x:0:root,HMUST
```

```

bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
:
:
```

Print the file with a specific indent of 8:

```

[ / ]
[ root@acme007 ][-sh]: cat /etc/group | indent 8
    root:x:0:root,HMUST
    bin:x:1:root,bin,daemon
    daemon:x:2:root,bin,daemon
    sys:x:3:root,bin,adm
    :
    :
```

2) indented output in a script

In the following script produces an output as:

```

:
:
interface files ...
    csv files in '/data/incoming' ...
        /data/incoming/HR/12002.csv
        /data/incoming/HR/12003.csv
        /data/incoming/HR/12004.csv
        /data/incoming/HR/12005.csv
        /data/incoming/GL/50003.csv
        /data/incoming/GL/50004.csv
    done.
    csv files in '/data/outgoing' ...
        /data/outgoing/HR/12001.csv
        /data/outgoing/GL/50001.csv
        /data/outgoing/GL/50002.csv
    done.
done.
:
:
```

Be aware of the 'cascaded' usage of the **indent** command first to indent the output of the **find** command and then to indent the output of the whole **search_csv()** function.

To ensure, that error output of the **find** command output is also indented, the output of **find** to **stderr** has to be redirected to **stdout** using the **2>&1** redirection syntax.

```

#!/bin/sh
:
:
search_csv(){ start=$1
    echo "csv files in '$start' ..."
}
```

```

        find $start -name *.csv -print 2>&1 | indent
        echo "done."
    } # search_csv

main() {

    in=/data/incoming
    out=/data/outgoing
    :
    :
    echo "interface files ..."
    search_csv $in | indent
    search_csv $out | indent
    echo "done."
    :
    :
} # main
main $*
```

The **main()** function above could also be written as:

```

main() {

    in=/data/incoming
    out=/data/outgoing
    :
    :
    echo "interface files ..."
    {
        search_csv $in
        search_csv $out
    } | indent
    echo "done."
    :
    :
} # main
```

SEE ALSO

edrcintro(1), **cat(1)**, **csv(3)**, **sh(1)**

NOTES

-

BUGS

-

AUTHOR

indent was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

info – read Info documents

SYNOPSIS

info [*OPTION*]... [*MENU-ITEM*...]

DESCRIPTION

Read documentation in Info format.

OPTIONS

- apropos=***SUBJECT*
look up *SUBJECT* in all indices of all manuals.
- directory=***DIR*
add *DIR* to INFOPATH.
- dribble=***FILENAME*
remember user keystrokes in *FILENAME*.
- file=***FILENAME*
specify Info file to visit.
- help** display this help and exit.
- index-search=***STRING*
go to node pointed by index entry *STRING*.
- node=***NODENAME*
specify nodes in first visited Info file.
- output=***FILENAME*
output selected nodes to *FILENAME*.
- restore=***FILENAME*
read initial keystrokes from *FILENAME*.
- show-options, --usage**
go to command-line options node.
- subnodes**
recursively output menu items.
- vi-keys**
use vi-like and less-like key bindings.
- version**
display version information and exit.

The first non-option argument, if present, is the menu entry to start from; it is searched for in all ‘dir’ files along INFOPATH. If it is not present, info merges all ‘dir’ files and shows the result. Any remaining arguments are treated as the names of menu items relative to the initial node visited.

EXAMPLES

- info show top-level dir menu
- info emacs
start at emacs node from top-level dir
- info emacs buffers
start at buffers node within emacs manual
- info **--show-options** emacs
start at node with emacs’ command line options
- info **-f** ./foo.info
show file ./foo.info, not searching dir

REPORTING BUGS

Email bug reports to bug-texinfo@gnu.org, general questions and discussion to help-texinfo@gnu.org.

COPYRIGHT

Copyright © 1999 Free Software Foundation, Inc. There is NO warranty. You may redistribute this software under the terms of the GNU General Public License. For more information about these matters, see the files named COPYING.

NAME

ini – initialize environment

SYNOPSIS

ini ([-h] | -l | *application*)

edrc/lib/ini.(bash|ksh|sh|csh) ([-h] | -l | *application*)

AVAILABILITY

WA2L/edrc

DESCRIPTION

initialize the environment based on named definitions specified in **etc/env**.

Each shell has its own **ini.shell** binary to initialize the environment based on the definitions.

So it is not necessary to use different syntax for different shells, this is done by the **ini.shell** command.

Interactive usage

The interactive **ini** command is a shell alias pointing to the relating **ini.shell** command.

The alias in the **ksh** shell therefore calls the **ini.ksh** command, a C-shell would call the **ini.csh** command.

Script usage

To use the **ini** mechanism in a script, invoke the **ini.shell** command and process the output:

Bourne-Shell using a temporary file and source it:

```
ini.sh myApplication > /tmp/env.$$  
. /tmp/env.$$
```

Korn-Shell using a temporary file and source it:

```
ini.ksh myApplication > /tmp/env.$$  
. /tmp/env.$$
```

Bourne-Again-Shell using a temporary file and source it:

```
ini.bash myApplication > /tmp/env.$$  
. /tmp/env.$$
```

C-Shell using a temporary file and source it:

```
ini.csh myApplication > /tmp/env.$$  
source /tmp/env.$$
```

Bourne-Shell using **eval**:

```
eval `ini.sh myApplication`
```

Korn-Shell using **eval**:

```
eval `ini.ksh myApplication`
```

Bourne-Again-Shell using **eval**:

```
eval `ini.bash myApplication`
```

C-Shell using **eval**:

```
eval `ini.csh myApplication`
```

OPTIONS

-h usage message.

-l list all application environment definitions from the **etc/env** configuration file.

application
 application name.

ENVIRONMENT

\$BASEDIR
 base directory of the userclass tool directory.

The userclass **env** files sit in: **\$BASEDIR/etc/**.

\$HOME users \$HOME directory. The env file is: **\$HOME/.myenv**.

EXIT STATUS

2 operating system not supported.

4 usage printed.

5 application definitions listed.

6 configuration file not found.

FILES**edrc/etc/ini.cfg**configuration file for **ini**.**edrc/etc/env**environment definition configuration file for **ini**.**\$HOME/.myenv**user specific environment definition configuration file for **ini**.**\$BASEDIR/etc/env**environment definition file if **USERCLASS_SUPPORT=True**.**\$BASEDIR/etc/env. domainname**domain dependent environment definition file if **USERCLASS_SUPPORT=True**.**\$BASEDIR/etc/env. hostname**host dependent environment definition file if **USERCLASS_SUPPORT=True**.**EXAMPLES**

-

SEE ALSO**edrcintro**(1), **env**(4), **ini.cfg**(4)**NOTES**

-

BUGS

-

AUTHORini was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .**COPYRIGHT**

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ini.cfg – configuration file for ini

SYNOPSIS

edrc/etc/ini.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **ini** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**USERCLASS_SUPPORT**

enable or disable USERCLASS support.

Example: USERCLASS_SUPPORT=True

Default: USERCLASS_SUPPORT=False

-

SEE ALSO

edrcintro(1), **env(4)**, **ini(1)**

NOTES

-

BUGS

-

AUTHOR

ini.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

input – prompt for an input

SYNOPSIS

edrc/lib/input

input "*prompt*" ["*default input*" [*options*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

This command is used in scripts to prompt a user for an input.

It is possible to verify the users input using a number of options. As long as the input does not conform to the option specified the input prompt is repeated.

It is also possible to specify a default value that takes in effect when <ENTER> or <RETURN> without entering an input is pressed.

All input is logged to a logfile if the environment variable **\$EDRC_LOGFILE** is set and the **NO_LOG** option isn't set.

OPTIONS

usage message

"*prompt*" prompt to be displayed.

"*default input*"

default input. This input will be returned if the <ENTER> key is pressed without entering anything on the input prompt.

options a list of comma separated options of the list below. The input is verified against all options listed. Therefore be careful not to provide an option list that makes it impossible to enter any data (e.g. **ALPHA,DIGIT**).

NOT_NULL

null input is not accepted.

NO_LOG

do not record the input to the logfile.

NO_TRIM

do not trim whitespaces from begin and end of input. Default is to trim those whitespaces.

Hint: If the input is a whitespace and the option **NOT_NULL** is set, the input is not accepted.

NO_ECHO

do not echo the input to stdout. This is useful if you query passwords.

LOG_STARS

log stars (*****) to the logfile instead of the real input. This option should be used always if you query passwords with the **input** command.

TTY read input from **tty** and write prompt to **tty**.

MINLEN=*len*

accept only input with a minimal length of *len*.

MAXLEN=*len*

accept only input up to a maximal length of *len*.

ALPHA

accept only inputs consisting of letters (a – Z).

DIGIT accept only numbers as unsigned integer values (0 – 9).

FLOAT

accept only numbers as signed floating point values. Examples: +.125, +0.125, 0.125, 3.1415, 815, -.125, -0.125, -3.1415, -815.

ALNUM

accept only alphanumeric input (a – Z, 0 – 9).

ALNUM+SPACE

accept only alphanumeric input including spaces (a – Z, 0 – 9, ' ').

ALNUM+_

accept only alphanumeric and _ (a – Z, 0 – 9, _).

IPV4 accept only IP version 4 addresses (0.0.0.0 – 255.255.255.255).

HH:MM

accept only time values in the format HH:MM (00:00 – 24:00).

HH:MM:SS

accept only time values in the format HH:MM:SS (00:00:00 – 24:00:00).

HHH:MM

accept only time (hour:minutes) values in the format hour:MM (00:00 – 9999999:59) in contrast to HH:MM, the hours are not limited to 24 hours.

HHH:MM:SS

accept only time values in the format hour:MM:SS (00:00:00 – 9999999:59:59) in contrast to HH:MM:SS, the hours are not limited to 24 hours.

MM/DD

accept only dates (month and day) in the format MM/DD (01/01 – 12/31).

MM/DD/YYYY

accept only dates in the format MM/DD/YYYY (01/01/0000 – 12/31/9999).

DD.MM

accept only dates (day and month) in the format DD.MM (01.01 – 31.12).

DD.MM.YYYY

accept only dates in the format DD.MM.YYYY (01.01.0000 – 31.12.9999).

YYYY-MM-DD

accept only dates in the military format YYYY-MM-DD (0000-01-01 – 9999-12-31).

ENVIRONMENT**\$EDRC_LOGFILE**

the accepted input read by **input** is logged to the file specified in this environment variable (if set).

EXIT STATUS

0 no error.

2 command aborted pressing Ctrl+C.

4 usage listed.

FILES

/dev/tty terminal device file when option **TTY** is used. This device file is read if the resolution of the current pseudo device of the terminal is not possible.

EXAMPLES

The following examples are script cut-outs of Bourne-, Korn- or Bash shell scripts:

1) filename input

```
filename=`input "Package Filename" "" NOT_NULL,ALNUM+_`
```

2) input a number and log everything to a logfile

```
Logfile=/var/log/process_counts.log
EDRC_LOGFILE=$Logfile; export EDRC_LOGFILE

pause=40
ok=n

{
    while [ ok != y ]; then
        pause=`input "pause between process counts" $pause DIGIT`
        ok=`choice "input OK? <yn>" yn n`
    done

    echo "press <Ctrl>+<C> to stop"
    while [ 1 = 1 ]; do
        echo "number of processes: `ps -ef | wc -l`"
        sleep $pause
    done
} >> $Logfile
```

3) prompt for a password input

```
Logfile=/var/log/passwordchange.log
EDRC_LOGFILE=$Logfile; export EDRC_LOGFILE

new=new
verify=verify

while [ "$new" != "$verify" ]; then
    new=`input "New password" "" NOT_NULL,LOG_STARS,NO_ECHO`
    verify=`input "Retype new password" "" NOT_NULL,LOG_STARS,NO_ECHO`
done
```

SEE ALSO

edrcintro(1), choice(3), input_targets(3), sh(1), ksh(1)

NOTES

-

BUGS

The dates 09/03/1752 - 09/13/1752 are accepted, despite of the fact that those dates did never occur because it is assumed that the Gregorian Reformation took place on the 09/03/1752.

AUTHOR

input was developed by Christian Walther. Send suggestions and bug reports to wa21@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

input_targets – input a list of target hosts and/or hostgroups

SYNOPSIS

edrc/lib/input_targets [*targets*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

special purpose input to query lists of hosts.

The prompt cannot be changed and is set to "Target hosts".

The input and the optional command argument is a space separated list of hosts and/or hostgroups (*@HOSTGRP*).

If hostgroups are entered, those hostgroups are resolved and the resulting list is a uniquely sorted list of target hosts. Therefore, if a host is member of one or more hostgroups and eventually entered as a separate target host, it will be listed only once in the resulting list.

If the input differs from the default, the input prompt is repeated to allow verification and modifications.

If no target list is specified in *targets*, a default target list is provided in the input prompt as it is resolved by the **hostlist**(3) command.

OPTIONS

targets space separated list of target hosts and/or hostgroups (*@HOSTGRP*).

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

input_targets(3)

Library Commands

input_targets(3)

EXAMPLES

-

SEE ALSO

edrcintro(1), **hostlist(3)**, **input(3)**, **resolve_targetlist(3)**

NOTES

-

BUGS

-

AUTHOR

input_targets was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ipcalc – IP Calculator

SYNOPSIS

edrc/bin/ipcalc [**--help**]

ipcalc [**options**] *address* [*/* *netmask*] [*netmask*]

ipcalc *address1-address2*

AVAILABILITY

WA2L/edrc

DESCRIPTION

ipcalc takes an IP address and netmask and calculates the resulting broadcast, network, Cisco wildcard mask, and host range. By giving a second netmask, you can design subnets and supernets. It is also intended to be a teaching tool and presents the subnetting results as easy-to-understand binary values.

Enter your netmask(s) in CIDR notation (*/25*) or dotted decimals (*255.255.255.0*). Inverse netmasks are recognized. If you omit the netmask **ipcalc** uses the default netmask for the class of your network.

Look at the space between the bits of the addresses: The bits before it are the network part of the address, the bits after it are the host part. You can see two simple facts: In a network address all host bits are zero, in a broadcast address they are all set.

The class of your network is determined by its first bits.

If your network is a private internet according to RFC 1918 this is remarked. When displaying subnets the new bits in the network part of the netmask are marked in a different color

The wildcard is the inverse netmask as used for access control lists in Cisco routers.

Do you want to split your network into subnets? Enter the address and netmask of your original network and play with the second netmask until the result matches your needs.

OPTIONS

--help usage message.

address IP-address.

netmask netmask.

- n** | **--nocolor**
do not display ANSI color codes.
- b** | **--nobinary**
suppress the bitwise output.
- c** | **--class** just print bit-count-mask of given address.
- h** | **--html** display results as HTML (not finished in this version).
- v** | **--version**
print Version.
- s** *n1 n2 n3*
split into networks of size n1, n2, n3.
- split** *n1 n2 n3*
split into networks of size n1, n2, n3.
- r** | **--range**
de-aggregate address range.

ENVIRONMENT

-

EXIT STATUS

0 always

FILES

-

EXAMPLES

1) common usage

```
ipcalc 192.168.0.1/24
ipcalc 192.168.0.1/255.255.128.0
ipcalc 192.168.0.1 255.255.128.0 255.255.192.0
ipcalc 192.168.0.1 0.0.63.255
```

2) de-aggregate address range

```
ipcalc 192.168.0.23 - 192.168.1.200
```

3) split networks into subnets

```
ipcalc 192.168.6.20 -s 200
ipcalc 192.168.6.20 -s 200 200 200
```

SEE ALSO

edrcintro(1), <http://www.ietf.org/rfc/rfc1517.txt>, http://www.cisco.com/univercd/cc/td/doc/product/software/ssr83/ptc_r/22057.htm, <https://wiki.archlinux.de/title/Ipcalc>

NOTES

The main part of this manpage has been extracted from the documentation of the **ipcalc** command as provided on the web site <http://jodies.de/ipcalc> of the author of the command, Krischan Jodies, and was modified to fit into WA2L/edrc package. Some of the examples come from the web page <https://wiki.archlinux.de/title/Ipcalc> which provides an excellent german description of the **ipcalc** command.

BUGS

-

AUTHOR

ipcalc was developed by Krischan Jodies <ipcalc-200808@jodies.de> and has been integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports related to WA2L/edrc to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ipsort – filter for sorted output of IP addresses

SYNOPSIS

edrc/bin/ipsort [-h]

AVAILABILITY

WA2L/edrc

DESCRIPTION

With this command you can sort a data stream which has IP addresses in the first column.

OPTIONS

-h usage message

EXAMPLES

1. Sorted output of the /etc/hosts file

```
myhost:[1] cat /etc/hosts | ipsort
```

2. Sorted output of the NIS hosts map

```
myhost:[2] ypcat hosts | ipsort
```

3. Sorted output of the actual routing table

```
myhost:[3] netstat -rnv | ipsort
```

SEE ALSO

edrcintro(1), **netstat(1M)**, **ypcat(1)**, **hosts(5)**

BUGS

-

AUTHOR

ipsort was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_config_byhand – check if file is listed in a .config_byhand* file

SYNOPSIS

edrc/lib/is_config_byhand [-h]

is_config_byhand -f file

AVAILABILITY

WA2L/edrc

DESCRIPTION

Evaluate if a given file is listed in a **config_byhand** file. If so return 'True' else return 'False'. This command is intended to be used in **pre_exec** and **post_exec** scripts called by **sysconfig** to prevent miss-configuration when re-applying a configuration to a previously configured system.

If **sysconfig** is called with the force option (-f), **is_sysconfig_byhand** returns *False* for all files queried.

OPTIONS

-h usage message.

-f file .

ENVIRONMENT**\$SYSCONFIG_FORCE**

if the environment variable **\$SYSCONFIG_FORCE** is set to *True* **is_config_byhand** returns *False* for every file. This environment variable is set to *True* when **sysconfig** is invoked with the force (-f) option.

EXIT STATUS

0 file is listed in a **config_byhand** file. **is_config_byhand** returns *True* to **stdout** in this case.

1 file is not listed in any **config_byhand** file. **is_config_byhand** returns *False* to **stdout** in this case.

2 operating system is not supported, yet. See **osid**(3) if you get this error.

- 4 usage printed.
- 11 temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

<root_dir>/EDRC.config_byhand <root_dir>/EDRC.config_byhand.<name>
a list of files to be protected from overwriting after the initial run of **sysconfig**.

<root_dir>/config_byhand
a list of files to be protected from overwriting after the initial run of **rmtconfig** a system configuration tool provided by the SFI-Director. You should not enter new entries into this file, it is only considered to protect systems configured with **rmtconfig** from miss-configuration.

EXAMPLES

1) usage in a post_exec script

Create a system dependent **/etc/issue** file.

```
Hostname='hostname'
Configuration='TEST CONFIGURATION - ACME, Datacenter Balsberg'
```

```
file=$SYSCONFIG_ROOTDIR/etc/issue
if [ "`is_config_byhand -f $file`" = False ]; then

    cat << EOM | apply2file -o -u bin -g bin -p 444 -f $file
```

```
`banner $Hostname`
```

```
$Configuration
```

```
#####
#   This is an ACME protected system.                               #
#   This system is for the use of authorized users only.           #
#   Individuals using this computer system without authority, or in #
#   excess of their authority, are subject to having all of their   #
#   activities on this system monitored and recorded by system     #
#   personnel.                                                       #
#                                                                     #
#   In the course of monitoring individuals improperly using this   #
#   system, or in the course of system maintenance, the activities #
#   of authorized users may also be monitored.                     #
#                                                                     #
#   Anyone using this system expressly consents to such monitoring  #
#   and is advised that if such monitoring reveals possible        #
#   evidence of criminal activity, system personnel may provide the #
#   evidence of such monitoring to law enforcement officials.       #
#####
```

```
EOM  
fi
```

SEE ALSO

apply2file(1), edrcintro(1), sysconfig(1)

NOTES

-

BUGS

-

AUTHOR

is_config_byhand was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_existing – test if a file or directory is existing

SYNOPSIS

edrc/lib/is_existing [**-h**]

is_existing *file* [*timeout*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

check if a *file/directory/link* etc. is existing and accessible by the calling user.

This command does not hang if for instance the NFS resource where the *file/directory/link* is located is not available.

There is the possibility to specify a timeout value when the default of 0 second is not sufficient.

OPTIONS

-h print usage message.

file file/directory/link to check if it exists.

timeout timeout in seconds.

This allows to check if a resource is existing that needs some time to get available (as an auto-mounted NFS directory).

the default timeout is 0 seconds.

ENVIRONMENT

-

EXIT STATUS

0 element exists. **True** is returned to **stdout**.

is_existing(3)

Library Commands

is_existing(3)

- 1** element is not existing or is not accessible within the given *timeout*. **False** is returned to **stdout**.
- 4** usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **is_writeable(3)**, **test(1)**

NOTES

-

BUGS

-

AUTHOR

is_existing was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_osid – check if this is a certain Os-Id

SYNOPSIS

edrc/lib/is_osid [**-h**]

is_osid [**-n**][**-s** *scriptname*] **-o** *osid_list*

is_osid [**-s** *scriptname*] **-o** *osid_list*

AVAILABILITY

WA2L/edrc

DESCRIPTION

evaluate if the operating system id returned by **osid**(3) matches to one os-id specified by the **-o** option. This command is used to limit a script execution to supported operating systems only.

OPTIONS

-h print usage message.

-n no message output, if the operating system id of the system where the **is_osid** command is running is not listed in the *osid_list*.

-s *scriptname*
scriptname to print in the message output. In a script this setting is normally set to **\$0**.

-o *osid_list*
comma separated list of **osid**(3) that are allowed to run the command.

ENVIRONMENT

-

EXIT STATUS

0 OS is in OS-Id list.

2 OS is not in Os-Id list (OS not supported).

4 usage printed.

FILES

-

EXAMPLES

1) check if osid is HP-11 or HP-11i

```
is_osid -s $0 -o HP-11,HP-11i || exit $?
```

2) check if osid is Solaris or HP-*:

```
is_osid -s $0 -o Solaris,HP-* || exit $?
```

SEE ALSO

edrcintro(1), **osid(3)**, **uname(1)**

NOTES

-

BUGS

-

AUTHOR

is_osid was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_permitted – check if a functionality is permitted to be started

SYNOPSIS

edrc/lib/is_permitted [**-h**]

is_permitted *functionality* [*options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

the main intention of this command is to check if a certain functionality is permitted for execution in contributed commands.

In recovery scripts global functionality and script functionality can be checked for startup permission.

This enables you to minimize possible by-passes of the denials defined in the **DENY_LIST** in the **edrc** config file **edrc.cfg**(4).

This allows to restrict functionality when an **edrc** configuration is used for special purposes (as: application- or operation support or to software control but don't want to give the user all freedom of changing things).

If a certain checked functionality (e.g.: */shell*) is not permitted to be started, the message

```
edrc-WARNING: execution of functionality 'shell' not permitted.
```

is displayed

OPTIONS

-h usage message.

functionality

functionality to check if it is permitted to start depending on the setting in the **DENY_LIST** of the **edrc** config file **edrc.cfg**(4).

The functionality is the string defined in the **@PROVIDES@** tag in the contributed command or general **edrc** functionalities.

To list all functionalities, invoke the **edrcperm** command in **edrc**.

Non-existing functionalities are treated as permitted.

Check if functionality of the own (e.g: edrcupgrade) contributed command is permitted to be started:

```
if [ `is_permitted seturl` = True ]; then
:
:
if
```

Check if global low-level **edrc** functionality (low level functionality has an underscore '_' in the name) is permitted to start:

```
if [ `is_permitted edit_file` = True ]; then
:
:
if
```

Check if a global **edrc** functionality is permitted to be started:

```
if [ `is_permitted /shell` = True ]; then
:
:
if

if [ `is_permitted /env.new` = True ]; then
:
:
if
```

Check if a specific functionality start of a specific contributed command is permitted:

```
if [ `is_permitted /contrib.logs.view` = True ]; then
:
:
if
```

options comma separated list or options for **is_permitted**:

NOLOG

do not log positive denial message into session log file.

ENVIRONMENT

-

EXIT STATUS

- 0** call of the functionality is permitted.
In addition **True** is returned to **stdout**.
- 1** call of the functionality is not permitted.
In addition **False** is returned to **stdout**.
- 4** usage displayed.

FILES

-

EXAMPLES

See contributed commands in **edrc/contrib/edrc/** for examples of the **is_permitted** command.

SEE ALSO

edrcintro(1), **contrib(1m)**, **contrib.edrc(1m)**, **edrc(1m)**, **edrc.cfg(4)**

NOTES

See also manpages **contrib(1m)**, **contrib.edrc(1m)**, **edrc(1m)** and **edrc.cfg(4)** to understand the **edrc** permission system.

BUGS

-

AUTHOR

is_permitted was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

is_permitted(3)

Library Commands

is_permitted(3)

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_running – check if a MC/Service Guard cluster package is running

SYNOPSIS

edrc/lib/is_running [**-h**]

is_running *package*

is_running *package*{*{,package_n}*}

AVAILABILITY

WA2L/edrc

DESCRIPTION

evaluate if a given cluster package is up.

Return 'True' if it is up, otherwise return nothing, therefore this command can be used in crontab to run a certain line only if the specified cluster package is running on this host.

Example:

```
0 1 * * * [ ``/opt/edrc/lib/is_running psoftprod`` ] \  
&& /opt/psoft/bin/my_command
```

OPTIONS

-h print usage message.

package MC/Service Guard cluster package name.

ENVIRONMENT

-

EXIT STATUS

0 one of the specified package(s) is up.

1 all specified package(s) are down.

2 OS not supported.

is_running(3)

Library Commands

is_running(3)

4 usage printed.

5 package state cannot be resolved.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1),

NOTES

-

BUGS

-

AUTHOR

is_running was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_up – test if a host is up

SYNOPSIS

edrc/lib/is_up -h

is_up *host*

AVAILABILITY

WA2L/edrc

DESCRIPTION

Test if a host is up by pinging it or by connecting to alternate ports on the system.

The advantage of using **is_up** instead of **ping**(8) and **ping6**(8) is, that this command has identical usage on different operating systems.

When **ping** (ICMP echo) is not allowed/enabled to test the peer's availability **is_up** tries to connect to the ports 22 and 513-514.

OPTIONS

-h print usage message.

host name or IP address of host to test.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|--|
| 0 | host reached. |
| 1 | host not reached. |
| 2 | operating system not supported. See osid (3) if you get this error. |
| 4 | usage message printed. |

is_up(3)

Library Commands

is_up(3)

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **nc**(3), **ping**(8), **ping6**(8), **portscan**(3), **rsh**(1), **ssh**(1)

NOTES

-

BUGS

-

AUTHOR

is_up was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_user – check if command is started by a certain user

SYNOPSIS

edrc/lib/is_user [**-h**]

is_user [**-n**] [**-s** *scriptname*] **-u** *user_list*

is_user [**-s** *scriptname*] **-u** *user_list*

AVAILABILITY

WA2L/edrc

DESCRIPTION

evaluate if the user that started the command matches to one username specified by the **-u** option. This command is used to limit a script execution to supported users only.

OPTIONS

-h print usage message.

-n no message output, if the username of the user starting the command where the **is_user** command is running is not listed in the *user_list*.

-s *scriptname*
scriptname to print in the message output. In a script this setting is normally set to **\$0**.

-u *user_list*
comma separated list of usernames that are allowed to run the command.

ENVIRONMENT

-

EXIT STATUS

0 User is in user-list list.

2 User is not in user-list (command start not permitted).

4 usage printed.

FILES

-

EXAMPLES

1) check if user is fred or barney

```
is_user -s $0 -u fred,barney || exit $?
```

2) check if user is root or admin*:

```
is_user -s $0 -u root,admin* || exit $?
```

SEE ALSO

edrcintro(1), **id(1)**, **is_osid(3)**

NOTES

-

BUGS

-

AUTHOR

is_user was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

is_weekend(3)

Library Commands

is_weekend(3)

NAME

is_weekend – return True on weekends, else False

SYNOPSIS

edrc/lib/is_weekend

AVAILABILITY

WA2L/edrc

DESCRIPTION

return **True** on weekends (Saturday and Sunday). Else **False** is returned.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 it is weekend (Saturday or Sunday) and **True** was returned to **stdout**

1 it is a working day (Monday ... Friday) and **False** was returned to **stdout**

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), date(1), days(3), seconds(3), timer(1), today(3), yesterday(3)

is_weekend(3)

Library Commands

is_weekend(3)

NOTES

-

BUGS

-

AUTHOR

is_weekend was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

is_writeable – test if a file or directory is writeable

SYNOPSIS

edrc/lib/is_writeable [-h]

is_writeable *file* | *directory*

AVAILABILITY

WA2L/edrc

DESCRIPTION

check if write access to a file or a directory is possible. Return 'True'/'0' if write access is possible, otherwise return 'False'/'1'.

OPTIONS

-h print usage message.

file file to check write access.

directory directory to check write access.

ENVIRONMENT

-

EXIT STATUS

0 write access possible. **True** is returned to **stdout**.

1 write access not possible. **False** is returned to **stdout**.

4 usage message printed.

FILES

-

is_writeable(3)

Library Commands

is_writeable(3)

EXAMPLES

-

SEE ALSO

edrcintro(1), **is_existing(3)**, **test(1)**

NOTES

-

BUGS

-

AUTHOR

is_writeable was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

java – start java

SYNOPSIS

edrc/bin/java [*java_options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

search the **java** command on the system and start it.

This command avoids the need to start the **java** command with an absolute path name and therefore allows to start a **java** program identically on different systems (for example from recovery scripts).

Locations of the **java** command can be configured in the **etc/java_wrapper.cfg** configuration file.

OPTIONS

java_options

all options of the system wide installed **java** command.

ENVIRONMENT

-

EXIT STATUS

101 the **java** command was not found on the system.

106 configuration file **etc/java_wrapper.cfg** does not exist.

x exit code of the started java program or the system wide installed **java** command.

FILES

etc/java_wrapper.cfg

configuration file of **java** and **.java_wrapper** .

EXAMPLES

-

SEE ALSO

edrcintro(1), **java_wrapper(1m)**, **java_wrapper.cfg(4)**

NOTES

Hint: the **java** runtime environment (JRE) or **java** development kit (JDK) is not (and will not be) bundled with WA2L/edrc.

BUGS

-

AUTHOR

java was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.
.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`java_wrapper` – wrap Java programs to ensure java startup

SYNOPSIS

`edrc/bin/javaprogram -> .java_wrapper`

`edrc/lib/javaprogram -> .java_wrapper`

AVAILABILITY

WA2L/edrc

DESCRIPTION

Wrap Java programs for transparent start.

java locations can be configured in the config file `java_wrapper.cfg`(4).

To start a new command thru the `.java_wrapper`, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ] [*edrc*/bash]: ln -s .java_wrapper new_cmd
```

- 2.) install the Java program in the **edrc/lib/java/** directory.
- 2.) define the Java start options in the related **edrc/lib/java/new_cmd.javaopt** file.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- | | |
|------------|---|
| 101 | shell (java) not found. If java is installed on the system and you get this error, add the java location in the configuration file <code>java_wrapper.cfg</code> . |
| 102 | the Java files to be started that should be located in lib/java/ do not exist. |

106 configuration file **java_wrapper.cfg** does not exist.

107 the **.java_wrapper** was called directly.

FILES

etc/java_wrapper.cfg

configuration file for the **java_wrapper** command.

lib/java/

location of the wrapped Java programs. This files should have the file permissions *644* to show, that those scripts should not be started directly.

EXAMPLES

-

SEE ALSO

edrcintro(1), **binprobe(1m)**, **daemon_wrapper(1)**, **java(1)**, **java_wrapper.cfg(4)**, **ksh_wrapper(1)**, **lua_wrapper(1)**, **ln(1)**, **ld(1)**, **osid(3)**, **os_wrapper(1)**, **perl_wrapper(1)**, **shlib(3)**

NOTES

-

BUGS

-

AUTHOR

java_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

java_wrapper.cfg – configuration file for .java_wrapper

SYNOPSIS

edrc/etc/java_wrapper.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **.java_wrapper** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**JAVA_COMMAND**

Space separated search sequence to the java command.

It is supported to use wildcards in the file path.

Example: JAVA_COMMAND="

```
/opt/java1.5*/jre/bin/java
/opt/java1.4*/jre/bin/java
/opt/prm/bin/jre/bin/java
/ora/product*/jre*/bin/java
/usr/bin/java
"
```

Default: JAVA_COMMAND="

```
/usr/java/jre*/bin/java
/usr/java/jdk*/bin/java
/usr/java/jdk*/jre/bin/java
edrc/lib/$OSID/jre*/bin/java
edrc/lib/$OSID/jdk*/bin/java
```

```
    /usr/bin/java  
"
```

SEE ALSO

edrcintro(1), **java_wrapper(1)**

NOTES

-

BUGS

-

AUTHOR

java_wrapper.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

job – job sequencer

SYNOPSIS

edrc/bin/job [**-h** | **-V**]

job

job [**-i** *check_interval*] [**-q** *queue*]

job (**-l** | **-L**) [**-q** *queue*]

job -Q

job -r (*id* | **last**)

job [**-f**] **-s** (*id* | **last**)

job -p (*id* | **last**)

job -a [**-q** *queue*]

job -c (*id* | **last**)

AVAILABILITY

WA2L/edrc

DESCRIPTION

job sequencer.

This command can be used to serialize jobs on a UNIX system within the environment of the current user (in the user context of the user).

It has a similar user interface to the **at** command.

To define a new job, start **job**, then add the commands to be executed interactively in the **job>** prompt.

To end adding commands to a job, press **[Ctrl] + [D]** or enter a . (dot) as last command.

The advantages of this command are multiple queue support, execution of jobs in the current user environment and a logging facility.

Be aware, that the directories **edrc/bin/** and **edrc/lib/** are added to the **\$PATH** environment variable and therefore all WA2L/edrc commands are also available for use in defined jobs without the need to alter the environment beforehand.

Background

The command was developed to optimize simulation calculations where only a number of licenses are available to simultaneously perform calculation runs.

The situation was, that a user needed to wait until the calculation of a coworker has been finished to start the own calculation.

This meant that the time during off-office hours were lost when a calculation ended after office hours.

Using the **job** sequencer the calculations could be en-queued and when one calculation finished, the following job started automatically. For each simultaneous calculation an own queue was used.

So the licenses could be used most efficiently and also night times were efficiently used to run calculations.

The job sequencer was developed in year 2000 as a stand alone utility and then later integrated into WA2L/edrc.

OPTIONS

- h** usage message.
- V** print program version.
- l** list all jobs in queue.
- L** list last jobs in queue.
- Q** list all queues.
- i interval** interval of check if previous job has completed.
- r id** remove job from queue.
- s id** stab job in run state.
- p id** print job properties.
- a** print *job* execution report for accounting purposes in **CSV** format.

To pretty-print the **CSV** output, use:

```
job -a | csv list
```

To select a specific project (e.g. APOLLO), the '**csv where *projectname***' command could be used:

```
job -a | csv where APOLLO | csv list
```

To show specific columns of the **CSV** output, the '**csv select *columns***' command could be used in addition:

```
job -a | csv where APOLLO | csv select "TAG;DURATION_S" | csv list
```

-c *id* print job commands including changing to the working directory from where the job was started.

This can be used to re-run a job using:

```
job -c 10123 | job
```

last resolves to the JOB-ID of the last job enqueued by the user.

-f force stabbing of job in run state with running processes.

-q *queue* queue name if not **default**.

ENVIRONMENT

\$PATH the directories **edrc/bin/** and **edrc/lib/** are added to the **\$PATH** environment variable.

Therefore all WA2L/edrc commands are available for use in jobs.

EXIT STATUS

| | |
|-----------|---|
| 0 | no error. |
| 4 | usage printed. |
| 5 | command aborted. |
| 6 | version printed. |
| 11 | cannot create temporary directory in /tmp/ ; this might be an indicator of system intrusion. |

FILES

etc/job.cfg configuration file for the **job**, **jobwatch**, and **joblog** commands.

lib/job/FUNCTION

functions that can be used in a **job**.

Currently available *FUNCTION*s are:

JOBBANNER *text*

print a banner (that is not limited to 10 characters).

JOBTITLE *text*

print a title with ### prepended.

JOBSUBTITLE *text*

print a sub-title with === prepended.

JOBSUBSUBTITLE *text*

print a sub-sub-title with --- prepended.

JOBTAG *tag*

tag a job for accounting. A *tag* could be a specific project name to identify which job was executed for what project.

Allowed characters for a *tag* are: **a-Z, 0-9, /, =, -, _, .** and **,**.

JOBCOMPLETED ["*command*"]

set the job completion criteria to *command*:

```
:
job>JOBCOMPLETED "/opt/mycalc/bin/checkflag"
:
```

whereas the *command* needs to return **0** when the job is completed and a value different to **0** when the job is not yet completed.

If the *command* is not set, the commands to check job completion need to be provided thru **stdin**, as for example:

```
:
job>JOBCOMPLETED <<EOC
job>    if [ -f /tmp/mycalc.flag ]; then
job>        exit 0
job>    else
job>        exit 1
job>    fi
job>EOC
:
```

The job completion criteria needs to be set in cases where for instance the commands defined in the job run in the background. In this case the **job** command would report the **done** state too early.

The **JOBCOMPLETED** function should be set prior to the job workload commands.

/var/tmp/job/job.log

log file of **job**.

/var/tmp/job/

spool directory of **job**.

/var/tmp/job.seq

job *id* counter.

~/job.last job *id* of the last job enqueued.

/var/tmp/job/<JOBID>/

spool directory of a job with the id <JOBID> consisting of the following state files:

cmd commands to run in the job.

env user environment (variables).

state state of the job.

pid process id of the job.

time.enqueue

time stamp when the job was created.

time.end time stamp when the job has ended.

time.run time stamp when commands in the job were started (when the job came into the run state).

exec job execution file.

user user that created the job.

cwd current working directory from where the **job** command was issued.

compl job completion criteria resolution.

log output (**stderr** and **stdout**) of the commands in the job.

version version of the **job** command at the time when the job was created.

queue job queue.

tag job tag.

EXAMPLES

-

SEE ALSO

edrcintro(1), **csv(3)**, **job.cfg(4)**, **joblog(1)**, **jobwatch(1)**

NOTES

-

BUGS

-

AUTHOR

job was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

job.cfg – configuration file for job/jobwatch/joblog

SYNOPSIS

edrc/etc/job.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **job**, **jobwatch** and **joblog** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SPOOLBASEDIR**

Location of the job spool base directory.

If this option is not set, the directory does not exist or is not writeable the **SPOOLBASEDIR** defaults to **/var/tmp**.

Example: SPOOLBASEDIR=/dat/simulator

Default: SPOOLBASEDIR=/var/tmp

COLORIZE

Colorize output of job list on terminal.

Example: COLORIZE=False

Default: COLORIZE=True

SEE ALSO

edrcintro(1), **job(1)**, **jobwatch(1)**, **joblog(1)**

NOTES

-

BUGS

-

AUTHOR

job.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

joblog – monitor log file of job sequencer

SYNOPSIS

edrc/bin/joblog [**-h** | **-V**]

joblog

joblog -j (*id* | **last**)

joblog -q [*queue*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

continuous output of the log file of the **job** sequencer.

If no *id* is specified, the main job sequencer log file is shown.

OPTIONS

-h usage message.

-V print program version.

-j *id* continuous output of the log file of the job with JOB-ID *id*.

last resolves to the last job enqueued.

-q [*queue*]
 permanent log output of **running** jobs in the given *queue*.

If the *queue* is not specified, the logs of the jobs in the **default** queue are printed.

ENVIRONMENT

-

EXIT STATUS

| | |
|-----------|---|
| 0 | no error. |
| 4 | usage printed. |
| 5 | command aborted. |
| 6 | version printed. |
| 11 | cannot create temporary directory in /tmp/ ; this might be an indicator of system intrusion. |

FILES

etc/job.cfg configuration file for the **job**, **jobwatch**, and **joblog** commands.

/var/tmp/job/job.log
main log file of **job**.

/var/tmp/job/<id>/log
log file of the job with JOB-ID *id*.

EXAMPLES

-

SEE ALSO

edrcintro(1), **job(1)**, **job.cfg(4)**, **jobwatch(1)**

NOTES

-

BUGS

-

AUTHOR

joblog was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

jobstart – run jobs thru a simple interface

SYNOPSIS

edrc/bin/jobstart [-h | -V

jobstart *JOBNAME*...

jobstart (-e | fun | distribute)

jobstart (-l | list | time | timeline) [*query*]

jobstart (runs | runs-days | stats | stats-days) [*query*]

jobstart (enable | disable | unlock | lock) *JOBNAME*...

jobstart (enable | disable | unlock | lock) all

jobstart print *JOBNAME*

jobstart log [*begin* [*end*]]

jobstart logswitch

AVAILABILITY

WA2L/edrc

DESCRIPTION

Start jobs using a simple interface, where a job is identified by a *JOBNAME*.

The **jobstart** command can be called manually as user **root** from the command line, but running it scheduled (i.e. from **cron** or **at**) is the main purpose.

The **jobstart** command can be started on any system where it is available. However, the *JOBNAME* is started from the system that is configured in the **JOBSTART_SYSTEM=system** setting (e.g. **JOBSTART_SYSTEM="root@acme301.acme.ch"**) in the **jobstart.cfg** configuration file.

If a job runs on a target system different to the **JOBSTART_SYSTEM**, the job needs to connect to the related target system (using **ssh-exec** for example).

Therefore the **JOBSTART_SYSTEM** acts like a hub to execute all jobs.

This to simplify jobs writing, understanding the job start mechanism and to consolidate the log output centrally.

The intention is to implement the job commands in the **jobstart.cfg** file to fulfill a specific task. The user then can start those (probably more complex) tasks using the *JOBNAME* without the need to remember command line options or dependencies.

The output of **stderr** and **stdout** is logged to the logfile, therefore it is not necessarily needed to write to a dedicated logfile from the job.

The *JOBNAME* should be in upper case and only the underscore (_) character should be used as separator.

Add additional jobs to the **jobstart.cfg** file between the

```
# v----- JOBS FOLLOW -----v #
:
:
# ^----- JOBS END -----^ #
```

markers.

See section **EXAMPLES** for example job definitions.

OPTIONS

-h usage message.

-V print configuration file name and configuration file version.

-e edit the **jobstart.cfg** file in **vi(1)** editor.

Running jobs are not affected by the editing of the **jobstart.cfg** file.

Be aware, that the configuration file is saved automatically before editing using the **vsav(1)** command internally. Therefore when a messed-up file was saved, or to check changes, the **vls(1)**, **vdiff(1)**, **vrestore(1)** etc. commands can be used to verify changes or revert to a previous version.

-l | list list all *JOBNAME*s currently defined sorted by *JOBNAME* including a short description, a reference to additional job documentation and informs if and when the *JOBNAME* is scheduled in the **crontab** of root.

The name, description and the job documentation reference is extracted from the configuration file if the job is defined as:

```
JOBNAME)      # description of the job
```

or

```
JOBNAME)      # description of the job / [REF]
```

in the **jobstart.cfg** file.

time list all *JOBNAME*s currently defined sorted by scheduled *HOURL* and *MINUTE*.

timeline list all *JOBNAME*s currently defined sorted by scheduled *HOURL* and *MINUTE* whereas all range- and list-definitions are expanded to it's planned scheduled time.

- query** wildcard expression to include only a subset of *JOBNAME*s in the output.
- print** print definition of job with given *JOBNAME*.
- fun** list special utility functions that can be used in job definitions.
- runs** overview of all jobs started yesterday and today. The report is sorted by the *END* time of the completed jobs.
- runs-days** overview of all jobs started the last number of *days*. The report is sorted by the *END* time of the completed jobs.
- runs-0** overview of all jobs started since begin. The report is sorted by the *END* time of the completed jobs.
- stats** statistics of all jobs started the last 90 days.
- Only job *RUNS* that are not skipped (due to **jobstart.cfg** configuration file syntax errors (*EXIT-CODE* = 2) or are disabled/locked (*ACTIVE* = **False**)) are *COUNTED* for job runtime statistics calculations.
- Starts of jobs that are not defined are not listed.
- stats-days** statistics of all jobs started the last number of *days*.
- stats-0** statistics of all jobs started since begin.
- log** [*begin* [*end*]]
print the **jobstart.log** file between the *begin* and *end* line number.
- distribute** distribute the **jobstart.cfg** file to the relevant nodes as defined in the **DISTRIBUTION=host_list** in the environment to keep it in sync after changes.
- The *host_list* is a comma separated list of hosts (as returned by the **hostlist -l** command).
- JOBNAME...*
start a job identified by the *JOBNAME*.
- A space separated list of *JOBNAME*s can be defined to execute several jobs in sequence. The

jobstart command stops executing a *JOBNAME* sequence as soon as a *JOBNAME* has an exit code different to **0** (success) and **98** (deactivated).

enable enable a given job with *JOBNAME* or **all** jobs to be started.

disable disable a given job with *JOBNAME* or **all** current jobs from being started.

A disabled *JOBNAME* is listed as *ACTIVE False* when listing all jobs using **jobstart list**.

unlock unlock a given job with *JOBNAME* or **all** jobs to be started.

lock lock a given job with *JOBNAME* or **all** current jobs from being started.

lock provides a second level of disabling jobs from being started beside of the **jobstart disable** *JOBNAME* command.

jobstart lock *JOBNAME* is used to rather permanently prevent a job from being started where **jobstart disable** *JOBNAME* is used to rather temporarily prevent a job from being started.

A locked *JOBNAME* is listed as *ACTIVE False* when listing all jobs using **jobstart list**.

logswitch move the contents of the current log file **jobstart.log** to **jobstart.<TIMESTAMP>.log**.

After a **logswitch** the **runs**, **stats** and **log** options can still be used to query information of all **jobstart** log files (the current **jobstart.log** and all available **jobstart.<TIMESTAMP>.log** files) without the need to know in which file the information is stored.

A log switch can significantly improve performance for queries of more recent events without losing information as a normal log rotate would.

The **jobstart logswitch** command should probably be scheduled (using **ecrontab**(1) for example).

ENVIRONMENT

\$EDRC_DEBUG_JOBSTART_JOBS

set this environment variable to **True** to switch on jobs debugging and to **False** to switch off debugging.

EXIT STATUS

0 no error.

1 error.

- 2** the configuration file has syntax errors. No jobs will be started if a syntax error is detected in the configuration file.
- 3** configuration file **edrc/etc/jobstart.cfg** not found.
- 4** usage displayed.
- 5** command aborted (using **Ctrl+C**).
- 6** config file version displayed.
- 7** program file distributed using the **distribute** option.
- 8** job activated using the **jobstart enable JOBNAME** or **jobstart unlock JOBNAME** command.
- 9** job deactivated using the **jobstart disable JOBNAME** or **jobstart lock JOBNAME** command.
- 10** utility functions that can be used in jobs listed.
- 11** list of jobs displayed using the **list**, **time** or **timeline** option.
- 12** the configuration file **jobstart.cfg** has been edited using the **-e** option.
- 13** log file **jobstart.log** printed using the **log** option.
- 14** *JOBNAME* definition printed.
- 16** command start not permitted. This error occurs, if **jobstart** command is started by a user different to the user defined in the optional **STARTUSER=user** setting, or the user has no permission to write to the **jobstart.log** file, the **jobstart.locked** file or the **jobstart.disabled** file.
- 98** started *JOBNAME* is deactivated (by **jobstart disable JOBNAME** or **jobstart lock JOBNAME**
- 99** supplied *JOBNAME* is not defined.

FILES

edrc/etc/jobstart.cfg

configuration file with jobs definitions.

edrc/etc/jobstart.lib

optional environment specific function library to be used in jobs.

This functions could also be added to the **jobstart.cfg** config file, but when defining the functions in **jobstart.lib** the functions are listed when **jobstart fun** is used and the comment format is:


```
# myfunction opt1 opt2 -- description
```

edrc/var/cache/jobstart/jobstart.cache
cached property data of switched log files.

edrc/var/log/jobstart.log
log file of **jobstart**.

edrc/var/settings/jobstart.disabled
list of disabled jobs. This file is maintained when using **jobstart** (**enable** | **disable**) *JOBNAME*.

edrc/var/settings/jobstart.locked
list of locked jobs. This file is maintained when using **jobstart** (**unlock** | **lock**) *JOBNAME*.

EXAMPLES

1) example job definitions

A job is a shell **case** structure, which is

```
JOBNAME)    # job description
            echo "job code goes here ..."
            ;;
```

or with a reference to additional job documentation:

```
JOBNAME)    # job description / [REF]
            echo "job code goes here ..."
            ;;
```

The example below shows the definition of the jobs *SYS2HTML* and *CRONTAB*:

```
# v----- JOBS FOLLOW -----v #
CRONTAB)    # list JOBSTART_SYSTEM crontab / crond(3)
            #
            echo "# ===== system root crontab ====="
            crontab -l
            echo "# ===== edrc root crontab ====="
            ecrontab -l
            set_exitcode 0
            ;;

SYS2HTML)   # system configuration HTML
            #
            collect='hostlist'
            outputdir=$REPORTDIR/SYS2HTML/$TODAY

            # job workload
            #
            mkdir $outputdir >/dev/null 2>&1; chmod 755 $outputdir
            sys2html -o $outputdir $collect
            set_exitcode 0
            ;;

# ^----- JOBS END -----^ #
```

2) list jobs

To list all defined jobs, output correlates to example 1):

```
[ / ]
[ root@acme301 ][*edrc*/bash]: jobstart -l
```

| ACTIVE | JOBNAME | CRON | HOURL | MINUTE | DAY | MONTH | WEEKDAY | QUEUE | DESCRIPTION | DOCREF |
|--------|----------|------|-------|--------|-----|-------|---------|---------|------------------------------|----------|
| True | CRONTAB | | | | | | | | list JOBSTART_SYSTEM crontab | crond(3) |
| True | PUBLISH | YES | 08 | 00 | * | * | 1,3 | Q02-J01 | publish documentation files | |
| True | SYS2HTML | YES | 12 | 00 | * | * | 7 | Q01-J01 | system configuration HTML | |
| True | WEBSYNC | YES | 12 | 00 | * | * | 7 | Q01-J02 | synchronize webserver data | [DLD] |

(2)

The scheduling information is resolved from the **ecrontab** and **crontab** definition on the **JOB-START_SYSTEM**:

```
[ / ]
[ root@acme301 ][*edrc*/bash]: ecrontab -l
```

```
#
# acme301:~edrc/var/spool/cron/root - root crontab
#
# [000] 06.07.2021 CWa Initial Version
#
```

```
SHELL=/bin/bash
PATH=~edrc/bin:/usr/bin:/bin
```

| #MINUTE | HOURL | DAY | MONTH | WEEKDAY | COMMAND |
|---------|--------|--------|--------|----------------|-------------------------|
| #(0-59) | (0-23) | (1-31) | (1-12) | (1=Mon, 7=Sun) | |
| 00 | 12 | * | * | 7 | jobrun SYS2HTML WEBSYNC |
| 00 | 08 | * | * | 1,3 | jobrun PUBLISH |

SEE ALSO

edrcintro(1), **crontab**(1), **ecrontab**(1), **crond**(3), **fit**(3), **hostlist**(3), **jobstart.cfg**(4), **jobstart.lib**(4), **fnmatch**(3), **sh**(1), **ssh-exec**(1), **usage**(1), **vdiff**(1), **vls**(1), **vrestore**(1), **vsav**(1)

NOTES

Beside the operating system commands, all commands provided by the WA2L/edrc package can be used.

To use **jobstart** in an application, create a symlink to the **edrc/bin/jobstart** command with a different name:

Example:

```
ln -s ~edrc/bin/jobstart /opt/myApp/bin/jobctl
```

The **FILES** of such a **jobstart** instance are: **/opt/myApp/etc/jobctl.cfg**, **/opt/myApp/etc/jobctl.lib**, **/opt/myApp/log/jobctl.log**, **/opt/myApp/var/jobctl.disabled**, **/opt/myApp/var/jobctl.locked**, and **edrc/var/cache/jobstart/jobctl.cache**.

The setting **JOBRUN_SYSTEM=system** defines the system from where the jobs are started for an instance with any name of **jobstart** (**jobctl** in this example), as the **JOBSTART_SYSTEM=system** does for **jobstart**.

BUGS

-

AUTHOR

jobstart was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see <https://www.gnu.org/licenses/gpl-2.0.html> for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

jobwatch – watch a job sequencer queue

SYNOPSIS

edrc/bin/jobwatch [**-h** | **-V**]

jobwatch [**-i** *update_interval*] [**-q** *queue* | **-Q**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

watch the state of all jobs in a given **queue**.

The output is similar to the **top**(1) operating system command.

OPTIONS

-h usage message.

-V print program version.

-i *update_interval*
 maximal interval of output list update.

The default is 60 seconds, but if a job state changes the list is updated immediately.

-q *queue* queuename if not **default**.

-Q watch status of all queues.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage printed.

- 5** command aborted.
- 6** version printed.
- 11** cannot create temporary directory in **/tmp/**; this might be an indicator of system intrusion.

FILES

etc/job.cfg configuration file for the **job**, **jobwatch**, and **joblog** commands.

/var/tmp/job/
spool directory of **job**.

/var/tmp/job/job.log
log file of **job**.

/var/tmp/job/.seq
job *id* counter.

/var/tmp/job/<JOBID>/
spool directory of a job with the id **<JOBID>** consisting on the following state files:

cmd commands to run in the job.

env user environment (variables).

state state of the job.

pid process id of the job.

time.enqueue
time stamp when the job was created.

time.end time stamp when the job has ended.

time.run time stamp when commands in the job were started (when the job came into the run state).

exec job execution file.

user user that created the job.

cwd current working directory from where the **job** command was issued.

compl job completion criteria resolution.

log output (**stderr** and **stdout**) of the commands in the job.

version version of the **job** command at the time when the job was created.

queue job queue.

EXAMPLES

-

SEE ALSO

edrcintro(1), **job**(1), **job.cfg**(4), **joblog**(1)

NOTES

-

BUGS

-

AUTHOR

jobwatch was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

jq – Command–line JSON processor

SYNOPSIS

jq [*options...*] *filter* [*files...*]

jq can transform JSON in various ways, by selecting, iterating, reducing and otherwise mangling JSON documents. For instance, running the command **jq 'map(.price) | add'** will take an array of JSON objects as input and return the sum of their "price" fields.

jq can accept text input as well, but by default, **jq** reads a stream of JSON entities (including numbers and other literals) from **stdin**. Whitespace is only needed to separate entities such as 1 and 2, and true and false. One or more *files* may be specified, in which case **jq** will read input from those instead.

The *options* are described in the *INVOKING JQ* section; they mostly concern input and output formatting. The *filter* is written in the jq language and specifies how to transform the input file or document.

FILTERS

A jq program is a "filter": it takes an input, and produces an output. There are a lot of builtin filters for extracting a particular field of an object, or converting a number to a string, or various other standard tasks.

Filters can be combined in various ways – you can pipe the output of one filter into another filter, or collect the output of a filter into an array.

Some filters produce multiple results, for instance there's one that produces all the elements of its input array. Piping that filter into a second runs the second filter for each element of the array. Generally, things that would be done with loops and iteration in other languages are just done by gluing filters together in jq.

It's important to remember that every filter has an input and an output. Even literals like "hello" or 42 are filters – they take an input but always produce the same literal as output. Operations that combine two filters, like addition, generally feed the same input to both and combine the results. So, you can implement an averaging filter as **add / length** – feeding the input array both to the **add** filter and the **length** filter and then performing the division.

But that's getting ahead of ourselves. :) Let's start with something simpler:

INVOKING JQ

jq filters run on a stream of JSON data. The input to jq is parsed as a sequence of whitespace-separated JSON values which are passed through the provided filter one at a time. The output(s) of the filter are written to standard out, again as a sequence of whitespace-separated JSON data.

Note: it is important to mind the shell's quoting rules. As a general rule it's best to always quote (with single-quote characters) the jq program, as too many characters with special meaning to jq are also shell meta-characters. For example, **jq "foo"** will fail on most Unix shells because that will be the same as **jq foo**, which will generally fail because **foo is not defined**. When using the Windows command shell (cmd.exe) it's best to use double quotes around your jq program when given on the command-line (instead of the **-f program-file** option), but then double-quotes in the jq program need backslash escaping.

You can affect how jq reads and writes its input and output using some command-line options:

- **--version:**
Output the jq version and exit with zero.
- **--seq:**
Use the **application/json-seq** MIME type scheme for separating JSON texts in jq's input and output. This means that an ASCII RS (record separator) character is printed before each value on output and an ASCII LF (line feed) is printed after every output. Input JSON texts that fail to parse are ignored (but warned about), discarding all subsequent input until the next RS. This more also parses the output of jq without the **--seq** option.
- **--stream:**

Parse the input in streaming fashion, outputting arrays of path and leaf values (scalars and empty arrays or empty objects). For example, `"a"` becomes `[[],"a"]`, and `[[],"a"],["b"]` becomes `[[0],[]], [[1],"a"]`, and `[[1,0],"b"]`.

This is useful for processing very large inputs. Use this in conjunction with filtering and the **reduce** and **foreach** syntax to reduce large inputs incrementally.

- **--slurp/-s:**
Instead of running the filter for each JSON object in the input, read the entire input stream into a large array and run the filter just once.
- **--raw-input/-R:**
Don't parse the input as JSON. Instead, each line of text is passed to the filter as a string. If combined with **--slurp**, then the entire input is passed to the filter as a single long string.
- **--null-input/-n:**
Don't read any input at all! Instead, the filter is run once using **null** as the input. This is useful when using jq as a simple calculator or to construct JSON data from scratch.
- **--compact-output / -c:**
By default, jq pretty-prints JSON output. Using this option will result in more compact output by instead putting each JSON object on a single line.
- **--tab:**
Use a tab for each indentation level instead of two spaces.
- **--indent n:**
Use the given number of spaces (no more than 8) for indentation.
- **--color-output / -C** and **--monochrome-output / -M:**
By default, jq outputs colored JSON if writing to a terminal. You can force it to produce color even if writing to a pipe or a file using **-C**, and disable color with **-M**.
- **--ascii-output / -a:**
jq usually outputs non-ASCII Unicode codepoints as UTF-8, even if the input specified them as escape sequences (like `"\u03bc"`). Using this option, you can force jq to produce pure ASCII output with every non-ASCII character replaced with the equivalent escape sequence.
- **--unbuffered**
Flush the output after each JSON object is printed (useful if you're piping a slow data source into jq and piping jq's output elsewhere).
- **--sort-keys / -S:**
Output the fields of each object with the keys in sorted order.
- **--raw-output / -r:**
With this option, if the filter's result is a string then it will be written directly to standard output rather than being formatted as a JSON string with quotes. This can be useful for making jq filters talk to non-JSON-based systems.
- **--join-output / -j:**
Like **-r** but jq won't print a newline after each output.
- **-f filename / --from-file filename:**
Read filter from the file rather than from a command line, like awk's **-f** option. You can also use `'#'` to make comments.

- **-Ldirectory / -L directory:**

Prepend **directory** to the search list for modules. If this option is used then no builtin search list is used. See the section on modules below.

- **-e / --exit-status:**

Sets the exit status of jq to 0 if the last output values was neither **false** nor **null**, 1 if the last output value was either **false** or **null**, or 4 if no valid result was ever produced. Normally jq exits with 2 if there was any usage problem or system error, 3 if there was a jq program compile error, or 0 if the jq program ran.

- **--arg name value:**

This option passes a value to the jq program as a predefined variable. If you run jq with **--arg foo bar**, then **\$foo** is available in the program and has the value **"bar"**. Note that **value** will be treated as a string, so **--arg foo 123** will bind **\$foo** to **"123"**.

- **--argjson name JSON-text:**

This option passes a JSON-encoded value to the jq program as a predefined variable. If you run jq with **--argjson foo 123**, then **\$foo** is available in the program and has the value **123**.

- **--slurpfile variable-name filename:**

This option reads all the JSON texts in the named file and binds an array of the parsed JSON values to the given global variable. If you run jq with **--argfile foo bar**, then **\$foo** is available in the program and has an array whose elements correspond to the texts in the file named **bar**.

- **--argfile variable-name filename:**

Do not use. Use **--slurpfile** instead.

(This option is like **--slurpfile**, but when the file has just one text, then that is used, else an array of texts is used as in **--slurpfile**.)

- **--run-tests [filename]:**

Runs the tests in the given file or standard input. This must be the last option given and does not honor all preceding options. The input consists of comment lines, empty lines, and program lines followed by one input line, as many lines of output as are expected (one per output), and a terminating empty line. Compilation failure tests start with a line containing only **"%%FAIL"**, then a line containing the program to compile, then a line containing an error message to compare to the actual.

Be warned that this option can change backwards-incompatibly.

BASIC FILTERS

•

The absolute simplest (and least interesting) filter is **..**. This is a filter that takes its input and produces it unchanged as output.

Since jq by default pretty-prints all output, this trivial program can be a useful way of formatting JSON output from, say, **curl**.

```
jq '
  "Hello, world!"
=> "Hello, world!"
```

.foo, .foo.bar

The simplest *useful* filter is **.foo**. When given a JSON object (aka dictionary or hash) as input, it produces the value at the key **"foo"**, or null if there's none present.

If the key contains special characters, you need to surround it with double quotes like this: `."foo$"`.

A filter of the form `.foo.bar` is equivalent to `.foo|.bar`.

```
jq '.foo'
{"foo": 42, "bar": "less interesting data"}
=> 42
```

```
jq '.foo'
{"notfoo": true, "alsonotfoo": false}
=> null
```

```
jq '["foo"]'
{"foo": 42}
=> 42
```

.foo?

Just like `.foo`, but does not output even an error when `.` is not an array or an object.

```
jq '.foo?'
{"foo": 42, "bar": "less interesting data"}
=> 42
```

```
jq '.foo?'
{"notfoo": true, "alsonotfoo": false}
=> null
```

```
jq '["foo"]?'
{"foo": 42}
=> 42
```

```
jq '[.foo?]'
[1,2]
=> []
```

.[<string>], .[2], .[10:15]

You can also look up fields of an object using syntax like `["foo"]` (`.foo` above is a shorthand version of this). This one works for arrays as well, if the key is an integer. Arrays are zero-based (like javascript), so `.[2]` returns the third element of the array.

The `.[10:15]` syntax can be used to return a subarray of an array or substring of a string. The array returned by `.[10:15]` will be of length 5, containing the elements from index 10 (inclusive) to index 15 (exclusive). Either index may be negative (in which case it counts backwards from the end of the array), or omitted (in which case it refers to the start or end of the array).

The `.[2]` syntax can be used to return the element at the given index. Negative indices are allowed, with `-1` referring to the last element, `-2` referring to the next to last element, and so on.

The `.foo` syntax only works for simply keys i.e. keys that are all alphanumeric characters. `.[<string>]` works with keys that contain special characters such as colons and dots. For example `["foo:bar"]` and `["foo.bar"]` work while `.foo::bar` and `.foo.bar` would not.

The `?` "operator" can also be used with the slice operator, as in `.[10:15]?`, which outputs values where the

inputs are slice-able.

```
jq '[0]'
[{"name":"JSON", "good":true}, {"name":"XML", "good":false}]
=> {"name":"JSON", "good":true}
```

```
jq '[2]'
[{"name":"JSON", "good":true}, {"name":"XML", "good":false}]
=> null
```

```
jq '[2:4]'
["a","b","c","d","e"]
=> ["c","d"]
```

```
jq '[2:4]'
"abcdefghi"
=> "cd"
```

```
jq '[:3]'
["a","b","c","d","e"]
=> ["a","b","c"]
```

```
jq '[-2:]'
["a","b","c","d","e"]
=> ["d","e"]
```

```
jq '[-2]'
[1,2,3]
=> 2
```

.[]

If you use the **.*[index]*** syntax, but omit the index entirely, it will return *all* of the elements of an array. Running **.[]** with the input **[1,2,3]** will produce the numbers as three separate results, rather than as a single array.

You can also use this on an object, and it will return all the values of the object.

```
jq '[]'
[{"name":"JSON", "good":true}, {"name":"XML", "good":false}]
=> {"name":"JSON", "good":true}, {"name":"XML", "good":false}
```

```
jq '[]'
[]
=>
```

```
jq '[]'
{"a": 1, "b": 1}
=> 1, 1
```

.[]?

Like **.[]**, but no errors will be output if **.** is not an array or object.

,

If two filters are separated by a comma, then the input will be fed into both and there will be multiple outputs: first, all of the outputs produced by the left expression, and then all of the outputs produced by the right. For instance, filter **.foo, .bar**, produces both the "foo" fields and "bar" fields as separate outputs.

```
jq ´.foo, .bar´
{"foo": 42, "bar": "something else", "baz": true}
=> 42, "something else"

jq ´.user, .projects[]´
{"user": "stedolan", "projects": ["jq", "wikiflow"]}
=> "stedolan", "jq", "wikiflow"

jq ´.[4,2]´
["a", "b", "c", "d", "e"]
=> "e", "c"
```

|

The **|** operator combines two filters by feeding the output(s) of the one on the left into the input of the one on the right. It's pretty much the same as the Unix shell's pipe, if you're used to that.

If the one on the left produces multiple results, the one on the right will be run for each of those results. So, the expression **.[] | .foo** retrieves the "foo" field of each element of the input array.

```
jq ´.[] | .name´
[{"name": "JSON", "good": true}, {"name": "XML", "good": false}]
=> "JSON", "XML"
```

TYPES AND VALUES

jq supports the same set of datatypes as JSON – numbers, strings, booleans, arrays, objects (which in JSON-speak are hashes with only string keys), and "null".

Booleans, null, strings and numbers are written the same way as in javascript. Just like everything else in jq, these simple values take an input and produce an output – **42** is a valid jq expression that takes an input, ignores it, and returns 42 instead.

Array construction – []

As in JSON, **[]** is used to construct arrays, as in **[1,2,3]**. The elements of the arrays can be any jq expression. All of the results produced by all of the expressions are collected into one big array. You can use it to construct an array out of a known quantity of values (as in **[.foo, .bar, .baz]**) or to "collect" all the results of a filter into an array (as in **[.items[].name]**)

Once you understand the **,** operator, you can look at jq's array syntax in a different light: the expression **[1,2,3]** is not using a built-in syntax for comma-separated arrays, but is instead applying the **[]** operator (collect results) to the expression **1,2,3** (which produces three different results).

If you have a filter **X** that produces four results, then the expression **[X]** will produce a single result, an array of four elements.

```
jq '[.user, .projects[]]'
  {"user":"stedolan", "projects": ["jq", "wikiflow"]}
=> ["stedolan", "jq", "wikiflow"]
```

Objects – {}

Like JSON, {} is for constructing objects (aka dictionaries or hashes), as in: {"a": 42, "b": 17}.

If the keys are "sensible" (all alphabetic characters), then the quotes can be left off. The value can be any expression (although you may need to wrap it in parentheses if it's a complicated one), which gets applied to the {} expression's input (remember, all filters have an input and an output).

```
{foo: .bar}
```

will produce the JSON object {"foo": 42} if given the JSON object {"bar":42, "baz":43}. You can use this to select particular fields of an object: if the input is an object with "user", "title", "id", and "content" fields and you just want "user" and "title", you can write

```
{user: .user, title: .title}
```

Because that's so common, there's a shortcut syntax: {user, title}.

If one of the expressions produces multiple results, multiple dictionaries will be produced. If the input's

```
{"user":"stedolan", "titles":["JQ Primer", "More JQ"]}
```

then the expression

```
{user, title: .titles[]}
```

will produce two outputs:

```
{"user":"stedolan", "title": "JQ Primer"}
{"user":"stedolan", "title": "More JQ"}
```

Putting parentheses around the key means it will be evaluated as an expression. With the same input as above,

```
{(.user): .titles}
```

produces

```

{"stedolan": ["JQ Primer", "More JQ"]}

jq `{"user, title: .titles[]}`
  {"user":"stedolan","titles":["JQ Primer", "More JQ"]}
=> {"user":"stedolan", "title": "JQ Primer"}, {"user":"stedolan", "title": "More JQ"}

jq `{(.user): .titles}`
  {"user":"stedolan","titles":["JQ Primer", "More JQ"]}
=> {"stedolan": ["JQ Primer", "More JQ"]}

```

BUILTIN OPERATORS AND FUNCTIONS

Some jq operator (for instance, `+`) do different things depending on the type of their arguments (arrays, numbers, etc.). However, jq never does implicit type conversions. If you try to add a string to an object you'll get an error message and no result.

Addition – `+`

The operator `+` takes two filters, applies them both to the same input, and adds the results together. What "adding" means depends on the types involved:

- **Numbers** are added by normal arithmetic.
- **Arrays** are added by being concatenated into a larger array.
- **Strings** are added by being joined into a larger string.
- **Objects** are added by merging, that is, inserting all the key–value pairs from both objects into a single combined object. If both objects contain a value for the same key, the object on the right of the `+` wins. (For recursive merge use the `*` operator.)

null can be added to any value, and returns the other value unchanged.

```

jq `a + 1`
  {"a": 7}
=> 8

jq `a + .b`
  {"a": [1,2], "b": [3,4]}
=> [1,2,3,4]

jq `a + null`
  {"a": 1}
=> 1

jq `a + 1`
  {}
=> 1

jq `{a: 1} + {b: 2} + {c: 3} + {a: 42}`
  null
=> {"a": 42, "b": 2, "c": 3}

```

Subtraction –

As well as normal arithmetic subtraction on numbers, the `–` operator can be used on arrays to remove all occurrences of the second array's elements from the first array.

```
jq '4 – .a'
{"a":3}
=> 1
```

```
jq '– ["xml", "yaml"]'
["xml", "yaml", "json"]
=> ["json"]
```

Multiplication, division, modulo – *, /, and %

These infix operators behave as expected when given two numbers. Division by zero raises an error. `x % y` computes x modulo y.

Multiplying a string by a number produces the concatenation of that string that many times. `"x" * 0` produces **null**.

Dividing a string by another splits the first using the second as separators.

Multiplying two objects will merge them recursively: this works like addition but if both objects contain a value for the same key, and the values are objects, the two are merged with the same strategy.

```
jq '10 / . * 3'
5
=> 6
```

```
jq '."/ ", "'
"a, b,c,d, e"
=> ["a","b,c,d","e"]
```

```
jq '{ "k": { "a": 1, "b": 2 } } * { "k": { "a": 0, "c": 3 } }'
null
=> { "k": { "a": 0, "b": 2, "c": 3 } }
```

```
jq '.[ ] | (1 / .)?'
[1,0,-1]
=> 1, -1
```

length

The builtin function **length** gets the length of various different types of value:

- The length of a **string** is the number of Unicode codepoints it contains (which will be the same as its JSON-encoded length in bytes if it's pure ASCII).
- The length of an **array** is the number of elements.
- The length of an **object** is the number of key–value pairs.
- The length of **null** is zero.

```
jq ´.[] | length´ [[1,2], "string", {"a":2}, null] => 2, 6, 1, 0
```

keys, keys_unsorted

The builtin function **keys**, when given an object, returns its keys in an array.

The keys are sorted "alphabetically", by unicode codepoint order. This is not an order that makes particular sense in any particular language, but you can count on it being the same for any two objects with the same set of keys, regardless of locale settings.

When **keys** is given an array, it returns the valid indices for that array: the integers from 0 to length-1.

The **keys_unsorted** function is just like **keys**, but if the input is an object then the keys will not be sorted, instead the keys will roughly be in insertion order.

```
jq ´keys´
  {"abc": 1, "abcd": 2, "Foo": 3}
=> ["Foo", "abc", "abcd"]
```

```
jq ´keys´
  [42,3,35]
=> [0,1,2]
```

has(key)

The builtin function **has** returns whether the input object has the given key, or the input array has an element at the given index.

has(\$key) has the same effect as checking whether **\$key** is a member of the array returned by **keys**, although **has** will be faster.

```
jq ´map(has("foo"))´
  [{"foo": 42}, {}]
=> [true, false]
```

```
jq ´map(has(2))´
  [[0,1], ["a","b","c"]]
=> [false, true]
```

in

The builtin function **in** returns the input key is in the given object, or the input index corresponds to an element in the given array. It is, essentially, an inversed version of **has**.

```
jq ´.[] | in({"foo": 42})´
  ["foo", "bar"]
=> true, false
```

```
jq ´map(in([0,1]))´
  [2, 0]
=> [false, true]
```


path(path_expression)

Outputs array representations of the given path expression in `..`. The outputs are arrays of strings (keys in objects) and/or numbers (array indices).

Path expressions are jq expressions like `.a`, but also `.[]`. There are two types of path expressions: ones that can match exactly, and ones that cannot. For example, `.a.b.c` is an exact match path expression, while `.a[].b` is not.

path(exact_path_expression) will produce the array representation of the path expression even if it does not exist in `..`, if `.` is **null** or an array or an object.

path(pattern) will produce array representations of the paths matching **pattern** if the paths exist in `..`.

Note that the path expressions are not different from normal expressions. The expression **path(..|select(type=="boolean"))** outputs all the paths to boolean values in `..`, and only those paths.

```
jq 'path(.a[0].b)'
null
=> ["a",0,"b"]

jq '[path(..)]'
{"a":[{"b":1}]}
=> [[],["a"],["a",0],["a",0,"b"]]
```

del(path_expression)

The builtin function **del** removes a key and its corresponding value from an object.

```
jq 'del(.foo)'
{"foo": 42, "bar": 9001, "baz": 42}
=> {"bar": 9001, "baz": 42}

jq 'del([1, 2])'
["foo", "bar", "baz"]
=> ["foo"]
```

to_entries, from_entries, with_entries

These functions convert between an object and an array of key–value pairs. If **to_entries** is passed an object, then for each **k: v** entry in the input, the output array includes **{"key": k, "value": v}**.

from_entries does the opposite conversion, and **with_entries(foo)** is a shorthand for **to_entries | map(foo) | from_entries**, useful for doing some operation to all keys and values of an object. **from_entries** accepts key, Key, Name, value and Value as keys.

```
jq 'to_entries'
{"a": 1, "b": 2}
=> [{"key": "a", "value": 1}, {"key": "b", "value": 2}]

jq 'from_entries'
[{"key": "a", "value": 1}, {"key": "b", "value": 2}]
=> {"a": 1, "b": 2}
```

```
jq 'with_entries(.key |= "KEY_" + .)'
  {"a": 1, "b": 2}
=> {"KEY_a": 1, "KEY_b": 2}
```

select(boolean_expression)

The function **select(foo)** produces its input unchanged if **foo** returns true for that input, and produces no output otherwise.

It's useful for filtering lists: **[1,2,3] | map(select(. >= 2))** will give you **[2,3]**.

```
jq 'map(select(. >= 2))'
  [1,5,3,0,7]
=> [5,3,7]

jq '.[] | select(.id == "second")'
  [{"id": "first", "val": 1}, {"id": "second", "val": 2}]
=> [{"id": "second", "val": 2}]
```

arrays, objects, iterables, booleans, numbers, normals, finites, strings, nulls, values, scalars

These built-ins select only inputs that are arrays, objects, iterables (arrays or objects), booleans, numbers, normal numbers, finite numbers, strings, null, non-null values, and non-iterables, respectively.

```
jq '.[]|numbers'
  [[],{}],1,"foo",null,true,false]
=> 1
```

empty

empty returns no results. None at all. Not even **null**.

It's useful on occasion. You'll know if you need it :)

```
jq '1, empty, 2'
  null
=> 1, 2

jq '[1,2,empty,3]'
  null
=> [1,2,3]
```

error(message)

Produces an error, just like **.a** applied to values other than null and objects would, but with the given message as the error's value.

\$_loc__

Produces an object with a "file" key and a "line" key, with the filename and line number where **\$_loc__** occurs, as values.

```
jq 'try error("\($__loc__)") catch .'
null
=> {"file\":"<top-level>","\line\:1}"
```

map(x), map_values(x)

For any filter **x**, **map(x)** will run that filter for each element of the input array, and produce the outputs a new array. **map(.+1)** will increment each element of an array of numbers.

Similarly, **map_values(x)** will run that filter for each element, but it will return an object when an object is passed.

map(x) is equivalent to `[.] | x`. In fact, this is how it's defined. Similarly, **map_values(x)** is defined as `[.] |= x`.

```
jq 'map(.+1)'
[1,2,3]
=> [2,3,4]

jq 'map_values(.+1)'
{"a": 1, "b": 2, "c": 3}
=> {"a": 2, "b": 3, "c": 4}
```

paths, paths(node_filter), leaf_paths

paths outputs the paths to all the elements in its input (except it does not output the empty list, representing itself).

paths(f) outputs the paths to any values for which **f** is true. That is, **paths(numbers)** outputs the paths to all numeric values.

leaf_paths is an alias of **paths(scalars)**; **leaf_paths** is *deprecated* and will be removed in the next major release.

```
jq '[paths]'
[1,[],{"a":2}]
=> [[0],[1],[1,0],[1,1],[1,1,"a"]]

jq '[paths(scalars)]'
[1,[],{"a":2}]
=> [[0],[1,1,"a"]]
```

add

The filter **add** takes as input an array, and produces as output the elements of the array added together. This might mean summed, concatenated or merged depending on the types of the elements of the input array – the rules are the same as those for the **+** operator (described above).

If the input is an empty array, **add** returns **null**.

```
jq 'add'
["a","b","c"]
=> "abc"
```

```
jq 'add'
  [1, 2, 3]
=> 6
```

```
jq 'add'
  []
=> null
```

any, any(condition), any(generator; condition)

The filter **any** takes as input an array of boolean values, and produces **true** as output if any of the the elements of the array is **true**.

If the input is an empty array, **any** returns **false**.

The **any(condition)** form applies the given condition to the elements of the input array.

The **any(generator; condition)** form applies the given condition to all the outputs of the given generator.

```
jq 'any'
  [true, false]
=> true
```

```
jq 'any'
  [false, false]
=> false
```

```
jq 'any'
  []
=> false
```

all, all(condition), all(generator; condition)

The filter **all** takes as input an array of boolean values, and produces **true** as output if all of the the elements of the array are **true**.

The **all(condition)** form applies the given condition to the elements of the input array.

The **all(generator; condition)** form applies the given condition to all the outputs of the given generator.

If the input is an empty array, **all** returns **true**.

```
jq 'all'
  [true, false]
=> false
```

```
jq 'all'
  [true, true]
=> true
```

```
jq 'all'
  []
=> true
```

[Requires 1.5] flatten, flatten(depth)

The filter **flatten** takes as input an array of nested arrays, and produces a flat array in which all arrays inside the original array have been recursively replaced by their values. You can pass an argument to it to specify how many levels of nesting to flatten.

flatten(2) is like **flatten**, but going only up to two levels deep.

```
jq 'flatten'
  [1, [2], [[3]]]
=> [1, 2, 3]

jq 'flatten(1)'
  [1, [2], [[3]]]
=> [1, 2, [3]]

jq 'flatten'
  [[]]
=> []

jq 'flatten'
  [{ "foo": "bar" }, [{ "foo": "baz" }]]
=> [{ "foo": "bar" }, { "foo": "baz" }]
```

range(upto), range(from;upto) range(from;upto;by)

The **range** function produces a range of numbers. **range(4;10)** produces 6 numbers, from 4 (inclusive) to 10 (exclusive). The numbers are produced as separate outputs. Use **[range(4;10)]** to get a range as an array.

The one argument form generates numbers from 0 to the given number, with an increment of 1.

The two argument form generates numbers from **from** to **upto** with an increment of 1.

The three argument form generates numbers **from** to **upto** with an increment of **by**.

```
jq 'range(2;4)'
  null
=> 2, 3

jq '[range(2;4)]'
  null
=> [2,3]

jq '[range(4)]'
  null
=> [0,1,2,3]

jq '[range(0;10;3)]'
  null
=> [0,3,6,9]

jq '[range(0;10;-1)]'
  null
=> []
```

```
jq '[range(0;-5;-1)]'
null
=> [0,-1,-2,-3,-4]
```

floor

The **floor** function returns the floor of its numeric input.

```
jq 'floor'
3.14159
=> 3
```

sqrt

The **sqrt** function returns the square root of its numeric input.

```
jq 'sqrt'
9
=> 3
```

tonumber

The **tonumber** function parses its input as a number. It will convert correctly-formatted strings to their numeric equivalent, leave numbers alone, and give an error on all other input.

```
jq '[] | tonumber'
[1, "1"]
=> 1, 1
```

tostring

The **tostring** function prints its input as a string. Strings are left unchanged, and all other values are JSON-encoded.

```
jq '[] | tostring'
[1, "1", [1]]
=> "1", "1", "[1]"
```

type

The **type** function returns the type of its argument as a string, which is one of null, boolean, number, string, array or object.

```
jq 'map(type)'
[0, false, [], {}, null, "hello"]
=> ["number", "boolean", "array", "object", "null", "string"]
```

infinite, nan, isinfinite, isnan, isfinite, isnormal

Some arithmetic operations can yield infinities and "not a number" (NaN) values. The **isinfinite** builtin returns **true** if its input is infinite. The **isnan** builtin returns **true** if its input is a NaN. The **infinite** builtin returns a positive infinite value. The **nan** builtin returns a NaN. The **isnormal** builtin returns true if its input is a normal number.

Note that division by zero raises an error.

Currently most arithmetic operations operating on infinities, NaNs, and sub-normals do not raise errors.

```
jq '[.] | (infinite * .) < 0'
  [-1, 1]
=> true, false

jq 'infinite, nan | type'
  null
=> "number", "number"
```

sort, sort_by(path_expression)

The **sort** functions sorts its input, which must be an array. Values are sorted in the following order:

- **null**
- **false**
- **true**
- numbers
- strings, in alphabetical order (by unicode codepoint value)
- arrays, in lexical order
- objects

The ordering for objects is a little complex: first they're compared by comparing their sets of keys (as arrays in sorted order), and if their keys are equal then the values are compared key by key.

sort may be used to sort by a particular field of an object, or by applying any jq filter.

sort_by(foo) compares two elements by comparing the result of **foo** on each element.

```
jq 'sort'
  [8,3,null,6]
=> [null,3,6,8]

jq 'sort_by(.foo)'
  [{"foo":4, "bar":10}, {"foo":3, "bar":100}, {"foo":2, "bar":1}]
=> [{"foo":2, "bar":1}, {"foo":3, "bar":100}, {"foo":4, "bar":10}]
```

group_by(path_expression)

group_by(.foo) takes as input an array, groups the elements having the same **.foo** field into separate arrays, and produces all of these arrays as elements of a larger array, sorted by the value of the **.foo** field.

Any jq expression, not just a field access, may be used in place of **.foo**. The sorting order is the same as

described in the **sort** function above.

```
jq 'group_by(.foo)'
[{"foo":1, "bar":10}, {"foo":3, "bar":100}, {"foo":1, "bar":1}]
=> [{"foo":1, "bar":10}, {"foo":1, "bar":1}], [{"foo":3, "bar":100}]
```

min, max, min_by(path_exp), max_by(path_exp)

Find the minimum or maximum element of the input array.

The **min_by(path_exp)** and **max_by(path_exp)** functions allow you to specify a particular field or property to examine, e.g. **min_by(.foo)** finds the object with the smallest **foo** field.

```
jq 'min'
[5,4,2,7]
=> 2

jq 'max_by(.foo)'
[{"foo":1, "bar":14}, {"foo":2, "bar":3}]
=> {"foo":2, "bar":3}
```

unique, unique_by(path_exp)

The **unique** function takes as input an array and produces an array of the same elements, in sorted order, with duplicates removed.

The **unique_by(path_exp)** function will keep only one element for each value obtained by applying the argument. Think of it as making an array by taking one element out of every group produced by **group**.

```
jq 'unique'
[1,2,5,3,5,3,1,3]
=> [1,2,3,5]

jq 'unique_by(.foo)'
[{"foo": 1, "bar": 2}, {"foo": 1, "bar": 3}, {"foo": 4, "bar": 5}]
=> [{"foo": 1, "bar": 2}, {"foo": 4, "bar": 5}]

jq 'unique_by(length)'
["chunky", "bacon", "kitten", "cicada", "asparagus"]
=> ["bacon", "chunky", "asparagus"]
```

reverse

This function reverses an array.

```
jq 'reverse'
[1,2,3,4]
=> [4,3,2,1]
```


contains(element)

The filter **contains(b)** will produce true if b is completely contained within the input. A string B is contained in a string A if B is a substring of A. An array B is contained in an array A if all elements in B are contained in any element in A. An object B is contained in object A if all of the values in B are contained in the value in A with the same key. All other types are assumed to be contained in each other if they are equal.

```
jq 'contains("bar")'
  "foobar"
=> true

jq 'contains(["baz", "bar"])'
  ["foobar", "foobaz", "blarp"]
=> true

jq 'contains(["bazzzz", "bar"])'
  ["foobar", "foobaz", "blarp"]
=> false

jq 'contains({foo: 12, bar: [{barp: 12}]})'
  {"foo": 12, "bar": [1,2,{"barp":12, "blip":13}]}
=> true

jq 'contains({foo: 12, bar: [{barp: 15}]})'
  {"foo": 12, "bar": [1,2,{"barp":12, "blip":13}]}
=> false
```

indices(s)

Outputs an array containing the indices in . where s occurs. The input may be an array, in which case if s is an array then the indices output will be those where all elements in . match those of s.

```
jq 'indices(",")'
  "a,b, cd, efg, hijk"
=> [3,7,12]

jq 'indices(1)'
  [0,1,2,1,3,1,4]
=> [1,3,5]

jq 'indices([1,2])'
  [0,1,2,3,1,4,2,5,1,2,6,7]
=> [1,8]
```

index(s), rindex(s)

Outputs the index of the first (**index**) or last (**rindex**) occurrence of s in the input.

```
jq 'index(",")'
```

```
"a,b, cd, efg, hijk"
=> 3
```

```
jq `rindex(", ")`
"a,b, cd, efg, hijk"
=> 12
```

inside

The filter **inside(b)** will produce true if the input is completely contained within b. It is, essentially, an inversed version of **contains**.

```
jq `inside("foobar")`
"bar"
=> true
```

```
jq `inside(["foobar", "foobaz", "blarp"])`
["baz", "bar"]
=> true
```

```
jq `inside(["foobar", "foobaz", "blarp"])`
["bazzzzz", "bar"]
=> false
```

```
jq `inside({"foo": 12, "bar": [1,2,{"barp":12, "blip":13}]})`
{"foo": 12, "bar": [{"barp": 12}]}
=> true
```

```
jq `inside({"foo": 12, "bar": [1,2,{"barp":12, "blip":13}]})`
{"foo": 12, "bar": [{"barp": 15}]}
=> false
```

startswith(str)

Outputs **true** if . starts with the given string argument.

```
jq `[.[]]|startswith("foo")`
["fo", "foo", "barfoo", "foobar", "barfoob"]
=> [false, true, false, true, false]
```

endswith(str)

Outputs **true** if . ends with the given string argument.

```
jq `[.[]]|endswith("foo")`
["foobar", "barfoo"]
=> [false, true]
```

combinations, combinations(n)

Outputs all combinations of the elements of the arrays in the input array. If given an argument **n**, it outputs all combinations of **n** repetitions of the input array.

```
jq 'combinations'
[[1,2], [3, 4]]
=> [1, 3], [1, 4], [2, 3], [2, 4]

jq 'combinations(2)'
[0, 1]
=> [0, 0], [0, 1], [1, 0], [1, 1]
```

ltrimstr(str)

Outputs its input with the given prefix string removed, if it starts with it.

```
jq '[.[]|ltrimstr("foo")]´
["fo", "foo", "barfoo", "foobar", "afoo"]
=> ["fo", "", "barfoo", "bar", "afoo"]
```

rtrimstr(str)

Outputs its input with the given suffix string removed, if it ends with it.

```
jq '[.[]|rtrimstr("foo")]´
["fo", "foo", "barfoo", "foobar", "foob"]
=> ["fo", "", "bar", "foobar", "foob"]
```

explode

Converts an input string into an array of the string's codepoint numbers.

```
jq 'explode'
"foobar"
=> [102,111,111,98,97,114]
```

implode

The inverse of explode.

```
jq 'implode'
[65, 66, 67]
=> "ABC"
```

split

Splits an input string on the separator argument.

```
jq 'split(", ")'
    "a, b,c,d, e, "
=> ["a","b,c,d","e",""]
```

join(str)

Joins the array of elements given as input, using the argument as separator. It is the inverse of **split**: that is, running **split("foo") | join("foo")** over any input string returns said input string.

```
jq 'join(", ")'
    ["a","b,c,d","e"]
=> "a, b,c,d, e"
```

ascii_lowercase, ascii_uppercase

Emit a copy of the input string with its alphabetic characters (a–z and A–Z) converted to the specified case.

while(cond; update)

The **while(cond; update)** function allows you to repeatedly apply an update to **.** until **cond** is false.

Note that **while(cond; update)** is internally defined as a recursive jq function. Recursive calls within **while** will not consume additional memory if **update** produces at most one output for each input. See advanced topics below.

```
jq '[while(<100; .*2)]'
    1
=> [1,2,4,8,16,32,64]
```

until(cond; next)

The **until(cond; next)** function allows you to repeatedly apply the expression **next**, initially to **.** then to its own output, until **cond** is true. For example, this can be used to implement a factorial function (see below).

Note that **until(cond; next)** is internally defined as a recursive jq function. Recursive calls within **until()** will not consume additional memory if **next** produces at most one output for each input. See advanced topics below.

```
jq '[.,1]|until(.[0] < 1; [.[0] - 1, .[1] * .[0]])|. [1]'
    4
=> 24
```

recurse(f), recurse, recurse(f; condition), recurse_down

The **recurse(f)** function allows you to search through a recursive structure, and extract interesting data from all levels. Suppose your input represents a filesystem:

```
{ "name": "/", "children": [
  { "name": "/bin", "children": [
    { "name": "/bin/l", "children": [] },
```

```
{ "name": "/bin/sh", "children": [] } },
{ "name": "/home", "children": [
  { "name": "/home/stephen", "children": [
    { "name": "/home/stephen/jq", "children": [] } ] } ] } }
```

Now suppose you want to extract all of the filenames present. You need to retrieve **.name**, **.children.name**, **.children[].children.name**, and so on. You can do this with:

```
recurse(children[]) | .name
```

When called without an argument, **recurse** is equivalent to **recurse(.[]?)**.

recurse(f) is identical to **recurse(f; . != null)** and can be used without concerns about recursion depth.

recurse(f; condition) is a generator which begins by emitting **.** and then emits in turn **.f**, **.f|f**, **.f|f|f**, ... so long as the computed value satisfies the condition. For example, to generate all the integers, at least in principle, one could write **recurse(+1; true)**.

For legacy reasons, **recurse_down** exists as an alias to calling **recurse** without arguments. This alias is considered *deprecated* and will be removed in the next major release.

The recursive calls in **recurse** will not consume additional memory whenever **f** produces at most a single output for each input.

```
jq 'recurse(.foo[])'
  { "foo": [{"foo": []}, {"foo": [{"foo": []}]} ] }
=> { "foo": [{"foo": []}, {"foo": [{"foo": []}]} ], {"foo": []}, {"foo": [{"foo": []}]} , {"foo": []}

jq 'recurse'
  { "a": 0, "b": [1] }
=> { "a": 0, "b": [1] }, 0, [1], 1

jq 'recurse(. * .; . < 20)'
  2
=> 2, 4, 16
```

..

Short-hand for **recurse** without arguments. This is intended to resemble the XPath **//** operator. Note that **..a** does not work; use **..|a** instead. In the example below we use **..|a?** to find all the values of object keys "a" in any object found "below" ..

```
jq '..|a?'
  [{"a": 1}]
=> 1
```

env

Outputs an object representing jq's environment.

```
jq 'env.PAGER'
null
=> "less"
```

transpose

Transpose a possibly jagged matrix (an array of arrays). Rows are padded with nulls so the result is always rectangular.

```
jq 'transpose'
[[1], [2,3]]
=> [[1,2],[null,3]]
```

bsearch(x)

bsearch(x) conducts a binary search for x in the input array. If the input is sorted and contains x, then bsearch(x) will return its index in the array; otherwise, if the array is sorted, it will return $(-1 - ix)$ where ix is an insertion point such that the array would still be sorted after the insertion of x at ix. If the array is not sorted, bsearch(x) will return an integer that is probably of no interest.

```
jq 'bsearch(0)'
[0,1]
=> 0
```

```
jq 'bsearch(0)'
[1,2,3]
=> -1
```

```
jq 'bsearch(4) as $ix | if $ix < 0 then .[-(1+$ix)] = 4 else . end'
[1,2,3]
=> [1,2,3,4]
```

String interpolation – \(\foo)

Inside a string, you can put an expression inside parens after a backslash. Whatever the expression returns will be interpolated into the string.

```
jq '"The input was \(.), which is one less than \(.+1)'"
42
=> "The input was 42, which is one less than 43"
```

Convert to/from JSON

The **tojson** and **fromjson** builtins dump values as JSON texts or parse JSON texts into values, respectively. The tojson builtin differs from tostring in that tostring returns strings unmodified, while tojson encodes strings as JSON strings.

```
jq '[.[]|tostring]'
[1, "foo", ["foo"]]
=> ["1","foo","\"foo\""]
```

```
jq '[.[]|tojson]`
  [1, "foo", ["foo"]]
=> [{"1","foo\""}, [{"foo\""}]
```

```
jq '[.[]|tojson|fromjson]`
  [1, "foo", ["foo"]]
=> [1,"foo",["foo"]]
```

Format strings and escaping

The **@foo** syntax is used to format and escape strings, which is useful for building URLs, documents in a language like HTML or XML, and so forth. **@foo** can be used as a filter on its own, the possible escapings are:

@text:

Calls **tostring**, see that function for details.

@json:

Serializes the input as JSON.

@html:

Applies HTML/XML escaping, by mapping the characters `<>&'"` to their entity equivalents **<**, **>**, **&**, **'**, **"**.

@uri:

Applies percent-encoding, by mapping all reserved URI characters to a **%XX** sequence.

@csv:

The input must be an array, and it is rendered as CSV with double quotes for strings, and quotes escaped by repetition.

@tsv:

The input must be an array, and it is rendered as TSV (tab-separated values). Each input array will be printed as a single line. Fields are separated by a single tab (ascii **0x09**). Input characters line-feed (ascii **0x0a**), carriage-return (ascii **0x0d**), tab (ascii **0x09**) and backslash (ascii **0x5c**) will be output as escape sequences **\n**, **\r**, **\t**, **** respectively.

@sh:

The input is escaped suitable for use in a command-line for a POSIX shell. If the input is an array, the output will be a series of space-separated strings.

@base64:

The input is converted to base64 as specified by RFC 4648.

This syntax can be combined with string interpolation in a useful way. You can follow a **@foo** token with a string literal. The contents of the string literal will *not* be escaped. However, all interpolations made inside that string literal will be escaped. For instance,

```
@uri "https://www.google.com/search?q=\\(.search)"
```

will produce the following output for the input **{"search":"what is jq?"}**:

```
"https://www.google.com/search?q=what%20is%20jq%3F"
```

Note that the slashes, question mark, etc. in the URL are not escaped, as they were part of the string literal.

```
jq '@html'
  "This works if x < y"
=> "This works if x &lt; y"

jq '@sh "echo \(.)"'
  "O'Hara's Ale"
=> "echo 'O\W'Hara\W's Ale'"
```

Dates

jq provides some basic date handling functionality, with some high-level and low-level builtins. In all cases these builtins deal exclusively with time in UTC.

The **fromdateiso8601** builtin parses datetimes in the ISO 8601 format to a number of seconds since the Unix epoch (1970-01-01T00:00:00Z). The **todateiso8601** builtin does the inverse.

The **fromdate** builtin parses datetime strings. Currently **fromdate** only supports ISO 8601 datetime strings, but in the future it will attempt to parse datetime strings in more formats.

The **todate** builtin is an alias for **todateiso8601**.

The **now** builtin outputs the current time, in seconds since the Unix epoch.

Low-level jq interfaces to the C-library time functions are also provided: **strptime**, **strftime**, **mktime**, and **gmtime**. Refer to your host operating system's documentation for the format strings used by **strptime** and **strftime**. Note: these are not necessarily stable interfaces in jq, particularly as to their localization functionality.

The **gmtime** builtin consumes a number of seconds since the Unix epoch and outputs a "broken down time" representation of time as an array of numbers representing (in this order): the year, the month (zero-based), the day of the month, the hour of the day, the minute of the hour, the second of the minute, the day of the week, and the day of the year — all one-based unless otherwise stated.

The **mktime** builtin consumes "broken down time" representations of time output by **gmtime** and **strptime**.

The **strptime(fmt)** builtin parses input strings matching the **fmt** argument. The output is in the "broken down time" representation consumed by **gmtime** and output by **mktime**.

The **strftime(fmt)** builtin formats a time with the given format.

The format strings for **strptime** and **strftime** are described in typical C library documentation. The format string for ISO 8601 datetime is "%Y-%m-%dT%H:%M:%SZ".

jq may not support some or all of this date functionality on some systems.

```
jq 'fromdate'
  "2015-03-05T23:51:47Z"
=> 1425599507

jq 'strptime("%Y-%m-%dT%H:%M:%SZ")'
  "2015-03-05T23:51:47Z"
=> [2015,2,5,23,51,47,4,63]

jq 'strptime("%Y-%m-%dT%H:%M:%SZ")|mktime'
  "2015-03-05T23:51:47Z"
```


=> 1425599507

CONDITIONALS AND COMPARISONS

==, !=

The expression `'a == b'` will produce `'true'` if the result of `a` and `b` are equal (that is, if they represent equivalent JSON documents) and `'false'` otherwise. In particular, strings are never considered equal to numbers. If you're coming from Javascript, jq's `==` is like Javascript's `===` – considering values equal only when they have the same type as well as the same value.

`!=` is "not equal", and `'a != b'` returns the opposite value of `'a == b'`

```
jq '.[] == 1'
[1, 1.0, "1", "banana"]
=> true, true, false, false
```

if-then-else

if A then B else C end will act the same as **B** if **A** produces a value other than false or null, but act the same as **C** otherwise.

Checking for false or null is a simpler notion of "truthiness" than is found in Javascript or Python, but it means that you'll sometimes have to be more explicit about the condition you want: you can't test whether, e.g. a string is empty using **if .name then A else B end**, you'll need something more like **if (.name | length) > 0 then A else B end** instead.

If the condition **A** produces multiple results, it is considered "true" if any of those results is not false or null. If it produces zero results, it's considered false.

More cases can be added to an if using **elif A then B** syntax.

```
jq 'if . == 0 then
```

```
"zero" elif . == 1 then "one" else "many" end' 2 => "many"
```

>, >=, <=, <

The comparison operators `>`, `>=`, `<=`, `<` return whether their left argument is greater than, greater than or equal to, less than or equal to or less than their right argument (respectively).

The ordering is the same as that described for **sort**, above.

```
jq '. < 5'
2
=> true
```

and/or/not

jq supports the normal Boolean operators and/or/not. They have the same standard of truth as if expressions – false and null are considered "false values", and anything else is a "true value".

If an operand of one of these operators produces multiple results, the operator itself will produce a result for each input.

not is in fact a builtin function rather than an operator, so it is called as a filter to which things can be piped rather than with special syntax, as in **.foo and .bar | not**.

These three only produce the values "true" and "false", and so are only useful for genuine Boolean operations, rather than the common Perl/Python/Ruby idiom of "value_that_may_be_null or default". If you want to use this form of "or", picking between two values rather than evaluating a condition, see the "///" operator below.

```
jq '42 and "a string"'
null
=> true

jq '(true, false) or false'
null
=> true, false

jq '(true, true) and (true, false)'
null
=> true, false, true, false

jq '[true, false | not]'
null
=> [false, true]
```

Alternative operator – //

A filter of the form **a // b** produces the same results as **a**, if **a** produces results other than **false** and **null**. Otherwise, **a // b** produces the same results as **b**.

This is useful for providing defaults: **.foo // 1** will evaluate to **1** if there's no **.foo** element in the input. It's similar to how **or** is sometimes used in Python (jq's **or** operator is reserved for strictly Boolean operations).

```
jq '.foo // 42'
{"foo": 19}
=> 19

jq '.foo // 42'
{}
=> 42
```

try-catch

Errors can be caught by using **try EXP catch EXP**. The first expression is executed, and if it fails then the second is executed with the error message. The output of the handler, if any, is output as if it had been the output of the expression to try.

The **try EXP** form uses **empty** as the exception handler.

```
jq 'try .a catch ". is not an object"'
true
=> ". is not an object"

jq '[.[]|try .a]'
[{}, true, {"a":1}]
=> [null, 1]
```

```
jq 'try error("some exception") catch .'
    true
=> "some exception"
```

Breaking out of control structures

A convenient use of try/catch is to break out of control structures like **reduce**, **foreach**, **while**, and so on.

For example:

```
# Repeat an expression until it raises "break" as an
# error, then stop repeating without re-raising the error.
# But if the error caught is not "break" then re-raise it.
try repeat(exp) catch .=="break" then empty else error;
```

jq has a syntax for named lexical labels to "break" or "go (back) to":

```
label $out | ... break $out ...
```

The **break \$label_name** expression will cause the program to act as though the nearest (to the left) **label \$label_name** produced **empty**.

The relationship between the **break** and corresponding **label** is lexical: the label has to be "visible" from the break.

To break out of a **reduce**, for example:

```
label $out | reduce .[] as $item (null; if .==false then break $out else ... end)
```

The following jq program produces a syntax error:

```
break $out
```

because no label **\$out** is visible.

? operator

The **?** operator, used as **EXP?**, is shorthand for **try EXP**.

```
jq '[.[]|(a)?]'
    [{}, true, {"a":1}]
=> [null, 1]
```

REGULAR EXPRESSIONS (PCRE)

jq uses the Oniguruma regular expression library, as do php, ruby, TextMate, Sublime Text, etc, so the description here will focus on jq specifics.

The jq regex filters are defined so that they can be used using one of these patterns:

```

STRING | FILTER( REGEX )
STRING | FILTER( REGEX; FLAGS )
STRING | FILTER( [REGEX] )
STRING | FILTER( [REGEX, FLAGS] )

```

where: * **STRING**, **REGEX** and **FLAGS** are jq strings and subject to jq string interpolation; * **REGEX**, after string interpolation, should be a valid PCRE regex; * **FILTER** is one of **test**, **match**, or **capture**, as described below.

FLAGS is a string consisting of one or more of the supported flags:

- **g** – Global search (find all matches, not just the first)
- **i** – Case insensitive search
- **m** – Multi line mode (‘.’ will match newlines)
- **n** – Ignore empty matches
- **p** – Both **s** and **m** modes are enabled
- **s** – Single line mode (‘^’ → ‘\A’, ‘\$’ → ‘\Z’)
- **l** – Find longest possible matches
- **x** – Extended regex format (ignore whitespace and comments)

To match whitespace in an **x** pattern use an escape such as `\s`, e.g.

- `test("a\s b", "x")`.

Note that certain flags may also be specified within **REGEX**, e.g.

- `jq -n '("test", "TEst", "teST", "TEST") | test("(?i)te(?-i)st")'`

evaluates to: true, true, false, false.

[Requires 1.5] **test(val)**, **test(regex; flags)**

Like **match**, but does not return match objects, only **true** or **false** for whether or not the regex matches the input.

```

jq 'test("foo")'
"foo"
=> true

```

```

jq '[.] | test("a b c # spaces are ignored"; "ix")'
["xabcd", "ABC"]
=> true, true

```

[Requires 1.5] **match(val)**, **match(regex; flags)**

match outputs an object for each match it finds. Matches have the following fields:

- **offset** – offset in UTF-8 codepoints from the beginning of the input

- **length** – length in UTF-8 codepoints of the match
- **string** – the string that it matched
- **captures** – an array of objects representing capturing groups.

Capturing group objects have the following fields:

- **offset** – offset in UTF-8 codepoints from the beginning of the input
- **length** – length in UTF-8 codepoints of this capturing group
- **string** – the string that was captured
- **name** – the name of the capturing group (or **null** if it was unnamed)

Capturing groups that did not match anything return an offset of `-1`

```
jq 'match("(abc)+"; "g")'
"abc abc"
=> { "offset": 0, "length": 3, "string": "abc", "captures": [{ "offset": 0, "length": 3, "string": "abc", "name": null }], { "offset": 3, "length": 3, "string": "abc", "name": null } }
```

```
jq 'match("foo")'
"foo bar foo"
=> { "offset": 0, "length": 3, "string": "foo", "captures": [] }
```

```
jq 'match(["foo", "ig"])'
"foo bar FOO"
=> { "offset": 0, "length": 3, "string": "foo", "captures": [] }, { "offset": 8, "length": 3, "string": "FOO", "captures": [] }
```

```
jq 'match("foo (?<bar123>bar)? foo"; "ig")'
"foo bar foo foo"
=> { "offset": 0, "length": 11, "string": "foo bar foo", "captures": [{ "offset": 4, "length": 3, "string": "bar", "name": "bar" }], { "offset": 11, "length": 3, "string": "foo", "name": null } }
```

```
jq '[ match(".", "g") ] | length'
"abc"
=> 3
```

[Requires 1.5] capture(val), capture(regex; flags)

Collects the named captures in a JSON object, with the name of each capture as the key, and the matched string as the corresponding value.

```
jq 'capture("(?<a>[a-z]+)-(?<n>[0-9]+)")'
"xyzy-14"
=> { "a": "xyzy", "n": "14" }
```

[Requires 1.5] scan(regex), scan(regex; flags)

Emit a stream of the non-overlapping substrings of the input that match the regex in accordance with the flags, if any have been specified. If there is no match, the stream is empty. To capture all the matches for each input string, use the idiom `[expr]`, e.g. `[scan(regex)]`.

split(regex; flags)

For backwards compatibility, **split** splits on a string, not a regex.

[Requires 1.5] splits(regex), splits(regex; flags)

These provide the same results as their **split** counterparts, but as a stream instead of an array.

[Requires 1.5] sub(regex; tostring) sub(regex; string; flags)

Emit the string obtained by replacing the first match of regex in the input string with **tostring**, after interpolation. **tostring** should be a jq string, and may contain references to named captures. The named captures are, in effect, presented as a JSON object (as constructed by **capture**) to **tostring**, so a reference to a captured variable named "x" would take the form: "(.x)".

[Requires 1.5] gsub(regex; string), gsub(regex; string; flags)

gsub is like **sub** but all the non-overlapping occurrences of the regex are replaced by the string, after interpolation.

ADVANCED FEATURES

Variables are an absolute necessity in most programming languages, but they're relegated to an "advanced feature" in jq.

In most languages, variables are the only means of passing around data. If you calculate a value, and you want to use it more than once, you'll need to store it in a variable. To pass a value to another part of the program, you'll need that part of the program to define a variable (as a function parameter, object member, or whatever) in which to place the data.

It is also possible to define functions in jq, although this is a feature whose biggest use is defining jq's standard library (many jq functions such as **map** and **find** are in fact written in jq).

jq has reduction operators, which are very powerful but a bit tricky. Again, these are mostly used internally, to define some useful bits of jq's standard library.

It may not be obvious at first, but jq is all about generators (yes, as often found in other languages). Some utilities are provided to help deal with generators.

Some minimal I/O support (besides reading JSON from standard input, and writing JSON to standard output) is available.

Finally, there is a module/library system.

Variables

In jq, all filters have an input and an output, so manual plumbing is not necessary to pass a value from one part of a program to the next. Many expressions, for instance **a + b**, pass their input to two distinct subexpressions (here **a** and **b** are both passed the same input), so variables aren't usually necessary in order to use a value twice.

For instance, calculating the average value of an array of numbers requires a few variables in most languages – at least one to hold the array, perhaps one for each element or for a loop counter. In jq, it's simply **add / length** – the **add** expression is given the array and produces its sum, and the **length** expression is given the array and produces its length.

So, there's generally a cleaner way to solve most problems in jq than defining variables. Still, sometimes they do make things easier, so jq lets you define variables using **expression as \$variable**. All variable names start with \$. Here's a slightly uglier version of the array-averaging example:

```
length as $array_length | add / $array_length
```

We'll need a more complicated problem to find a situation where using variables actually makes our lives easier.

Suppose we have an array of blog posts, with "author" and "title" fields, and another object which is used to map author usernames to real names. Our input looks like:

```
{ "posts": [{ "title": "Frist psot", "author": "anon"},
             { "title": "A well-written article", "author": "person1" }],
  "realnames": { "anon": "Anonymous Coward",
                 "person1": "Person McPherson" } }
```

We want to produce the posts with the author field containing a real name, as in:

```
{ "title": "Frist psot", "author": "Anonymous Coward" }
{ "title": "A well-written article", "author": "Person McPherson" }
```

We use a variable, `$names`, to store the `realnames` object, so that we can refer to it later when looking up author usernames:

```
.realnames as $names | .posts[] | {title, author: $names[.author]}
```

The expression **`exp as $x | ...`** means: for each value of expression **`exp`**, run the rest of the pipeline with the entire original input, and with **`$x`** set to that value. Thus **`as`** functions as something of a `foreach` loop.

Just as **`{foo}`** is a handy way of writing **`{foo: .foo}`**, so **`{foo}`** is a handy way of writing **`{foo:$foo}`**.

Multiple variables may be declared using a single **`as`** expression by providing a pattern that matches the structure of the input (this is known as "destructuring"):

```
. as {realnames: $names, posts: [$first, $second]} | ...
```

The variable declarations in array patterns (e.g., **`. as [$first, $second]`**) bind to the elements of the array in from the element at index zero on up, in order. When there is no value at the index for an array pattern element, **`null`** is bound to that variable.

Variables are scoped over the rest of the expression that defines them, so

```
.realnames as $names | (.posts[] | {title, author: $names[.author]})
```

will work, but

```
(.realnames as $names | .posts[]) | {title, author: $names[.author]}
```

won't.

For programming language theorists, it's more accurate to say that jq variables are lexically-scoped bindings. In particular there's no way to change the value of a binding; one can only setup a new binding with the same name, but which will not be visible where the old one was.

```
jq ´.bar as $x | .foo | . + $x´
{"foo":10, "bar":200}
=> 210

jq ´. as $i[[:.*2|. as $i| $i), $i]´
5
=> [10,5]

jq ´. as [$a, $b, {c: $c}] | $a + $b + $c´
[2, 3, {"c": 4, "d": 5}]
=> 9

jq ´.[] as [$a, $b] | {a: $a, b: $b}´
[[0], [0, 1], [2, 1, 0]]
=> {"a":0,"b":null}, {"a":0,"b":1}, {"a":2,"b":1}
```

Defining Functions

You can give a filter a name using "def" syntax:

```
def increment: . + 1;
```

From then on, **increment** is usable as a filter just like a builtin function (in fact, this is how some of the builtins are defined). A function may take arguments:

```
def map(f): [.] | f;
```

Arguments are passed as filters, not as values. The same argument may be referenced multiple times with different inputs (here **f** is run for each element of the input array). Arguments to a function work more like callbacks than like value arguments. This is important to understand. Consider:

```
def foo(f): f|f;
5|foo(*2)
```

The result will be 20 because **f** is ***2**, and during the first invocation of **f** . will be 5, and the second time it will be 10 (5 * 2), so the result will be 20. Function arguments are filters, and filters expect an input when invoked.

If you want the value–argument behaviour for defining simple functions, you can just use a variable:

```
def addvalue(f): f as $f | map(. + $f);
```

Or use the short–hand:


```
def addvalue($f): ...;
```

With either definition, **addvalue.foo** will add the current input's **.foo** field to each element of the array.

Multiple definitions using the same function name are allowed. Each re-definition replaces the previous one for the same number of function arguments, but only for references from functions (or main program) subsequent to the re-definition.

```
jq 'def addvalue(f): . + [f]; map(addvalue(.[]))'
  [[1,2],[10,20]]
=> [[1,2,1], [10,20,10]]

jq 'def addvalue(f): f as $x | map(. + $x); addvalue(.[])'
  [[1,2],[10,20]]
=> [[1,2,1,2], [10,20,1,2]]
```

Reduce

The **reduce** syntax in jq allows you to combine all of the results of an expression by accumulating them into a single answer. As an example, we'll pass **[3,2,1]** to this expression:

```
reduce .[] as $item (0; . + $item)
```

For each result that **.[]** produces, **. + \$item** is run to accumulate a running total, starting from 0. In this example, **.[]** produces the results 3, 2, and 1, so the effect is similar to running something like this:

```
0 | (3 as $item | . + $item) |
   (2 as $item | . + $item) |
   (1 as $item | . + $item)

jq 'reduce .[] as $item (0; . + $item)'
  [10,2,5,3]
=> 20
```

limit(n; exp)

The **limit** function extracts up to **n** outputs from **exp**.

```
jq '[limit(3;.)]'
  [0,1,2,3,4,5,6,7,8,9]
=> [0,1,2]
```

first(expr), last(expr), nth(n; expr)

The **first(expr)** and **last(expr)** functions extract the first and last values from **expr**, respectively.

The **nth(n; expr)** function extracts the **n**th value output by **expr**. This can be defined as **def nth(n; expr):**

last(limit(n + 1; expr)); Note that **nth(n; expr)** doesn't support negative values of **n**.

```
jq '[first(range(.)), last(range(.)), nth(. / 2; range(.))]'
10
=> [0,9,5]
```

first, last, nth(n)

The **first** and **last** functions extract the first and last values from any array at ..

The **nth(n)** function extracts the nth value of any array at ..

```
jq '[range(.)][first, last, nth(5)]'
10
=> [0,9,5]
```

foreach

The **foreach** syntax is similar to **reduce**, but intended to allow the construction of **limit** and reducers that produce intermediate results (see example).

The form is **foreach EXP as \$var (INIT; UPDATE; EXTRACT)**. Like **reduce**, **INIT** is evaluated once to produce a state value, then each output of **EXP** is bound to **\$var**, **UPDATE** is evaluated for each output of **EXP** with the current state and with **\$var** visible. Each value output by **UPDATE** replaces the previous state. Finally, **EXTRACT** is evaluated for each new state to extract an output of **foreach**.

This is mostly useful only for constructing **reduce**- and **limit**-like functions. But it is much more general, as it allows for partial reductions (see the example below).

```
jq 'foreach .[] as $item ([[],[]]; if $item == null then [[],[0]] else [(.[] + [$item]),[]] end; if $item == null then .[1] *
[1,2,3,4,null,"a","b",null]
=> [[1,2,3,4],["a","b"]]
```

Recursion

As described above, **recurse** uses recursion, and any jq function can be recursive. The **while** builtin is also implemented in terms of recursion.

Tail calls are optimized whenever the expression to the left of the recursive call outputs its last value. In practice this means that the expression to the left of the recursive call should not produce more than one output for each input.

For example:

```
def recurse(f): def r: ., (f | select(. != null) | r); r;

def while(cond; update):
  def _while:
    if cond then ., (update | _while) else empty end;
  _while;

def repeat(exp):
```

```
def _repeat:
  exp, _repeat;
_repeat;
```

Generators and iterators

Some jq operators and functions are actually generators in that they can produce zero, one, or more values for each input, just as one might expect in other programming languages that have generators. For example, `.[]` generates all the values in its input (which must be an array or an object), `range(0; 10)` generates the integers between 0 and 10, and so on.

Even the comma operator is a generator, generating first the values generated by the expression to the left of the comma, then for each of those, the values generated by the expression on the right of the comma.

The **empty** builtin is the generator that produces zero outputs. The **empty** builtin backtracks to the preceding generator expression.

All jq functions can be generators just by using builtin generators. It is also possible to define new generators using only recursion and the comma operator. If the recursive call(s) is(are) "in tail position" then the generator will be efficient. In the example below the recursive call by `_range` to itself is in tail position. The example shows off three advanced topics: tail recursion, generator construction, and sub-functions.

```
jq 'def range(init; upto; by): def _range: if (by > 0 and . < upto) or (by < 0 and . > upto) then ., ((.+by)|_range) else .;
  null
=> 0, 3, 6, 9

jq 'def while(cond; update): def _while: if cond then ., (update | _while) else empty end; _while; [while(<100; .*2)]'
1
=> [1,2,4,8,16,32,64]
```

MATH

jq currently only has IEEE754 double-precision (64-bit) floating point number support.

Besides simple arithmetic operators such as `+`, jq also has most standard math functions from the C math library. C math functions that take a single input argument (e.g., `sin()`) are available as zero-argument jq functions. C math functions that take two input arguments (e.g., `pow()`) are available as two-argument jq functions that ignore `..`.

Availability of standard math functions depends on the availability of the corresponding math functions in your operating system and C math library. Unavailable math functions will be defined but will raise an error.

I/O

At this time jq has minimal support for I/O, mostly in the form of control over when inputs are read. Two builtin functions are provided for this, **input** and **inputs**, that read from the same sources (e.g., `stdin`, files named on the command-line) as jq itself. These two builtins, and jq's own reading actions, can be interleaved with each other.

One builtin provides minimal output capabilities, **debug**. (Recall that a jq program's output values are always output as JSON texts on `stdout`.) The **debug** builtin can have application-specific behavior, such as for executables that use the libjq C API but aren't the jq executable itself.

input

Outputs one new input.

inputs

Outputs all remaining inputs, one by one.

This is primarily useful for reductions over a program's inputs.

debug

Causes a debug message based on the input value to be produced. The jq executable wraps the input value with ["**DEBUG:**", <input-value>] and prints that and a newline on stderr, compactly. This may change in the future.

input_filename

Returns the name of the file whose input is currently being filtered. Note that this will not work well unless jq is running in a UTF-8 locale.

input_line_number

Returns the line number of the input currently being filtered.

STREAMING

With the **--stream** option jq can parse input texts in a streaming fashion, allowing jq programs to start processing large JSON texts immediately rather than after the parse completes. If you have a single JSON text that is 1GB in size, streaming it will allow you to process it much more quickly.

However, streaming isn't easy to deal with as the jq program will have [<path>, <leaf-value>] (and a few other forms) as inputs.

Several builtins are provided to make handling streams easier.

The examples below use the the streamed form of [0,[1]], which is [[0],0],[[1,0],1],[[1,0]],[[1]].

Streaming forms include [<path>, <leaf-value>] (to indicate any scalar value, empty array, or empty object), and [<path>] (to indicate the end of an array or object). Future versions of jq run with **--stream** and **-seq** may output additional forms such as ["error message"] when an input text fails to parse.

truncate_stream(stream_expression)

Consumes a number as input and truncates the corresponding number of path elements from the left of the outputs of the given streaming expression.

```
jq '[1|truncate_stream([[0],1],[[1,0],2],[[1,0]],[[1]])]'
1
=> [[[0],2],[[0]]]
```

fromstream(stream_expression)

Outputs values corresponding to the stream expression's outputs.

```
jq 'fromstream(1|truncate_stream([[0],1],[[1,0],2],[[1,0]],[[1]]))'
null
=> [2]
```

tostream

The **tostream** builtin outputs the streamed form of its input.

```
jq '. as $dot|fromstream($dot|tostream)|==$dot'
[0,[1,{"a":1},{ "b":2}]]
=> true
```

ASSIGNMENT

Assignment works a little differently in jq than in most programming languages. jq doesn't distinguish between references to and copies of something – two objects or arrays are either equal or not equal, without any further notion of being "the same object" or "not the same object".

If an object has two fields which are arrays, **.foo** and **.bar**, and you append something to **.foo**, then **.bar** will not get bigger. Even if you've just set **.bar = .foo**. If you're used to programming in languages like Python, Java, Ruby, Javascript, etc. then you can think of it as though jq does a full deep copy of every object before it does the assignment (for performance, it doesn't actually do that, but that's the general idea).

All the assignment operators in jq have path expressions on the left-hand side.

=

The filter **.foo = 1** will take as input an object and produce as output an object with the "foo" field set to 1. There is no notion of "modifying" or "changing" something in jq – all jq values are immutable. For instance,

```
.foo = .bar | .foo.baz = 1
```

will not have the side-effect of setting **.bar.baz** to be set to 1, as the similar-looking program in Javascript, Python, Ruby or other languages would. Unlike these languages (but like Haskell and some other functional languages), there is no notion of two arrays or objects being "the same array" or "the same object". They can be equal, or not equal, but if we change one of them in no circumstances will the other change behind our backs.

This means that it's impossible to build circular values in jq (such as an array whose first element is itself). This is quite intentional, and ensures that anything a jq program can produce can be represented in JSON.

Note that the left-hand side of '=' refers to a value in **..**. Thus **\$var.foo = 1** won't work as expected (**\$var.foo** is not a valid or useful path expression in **..**); use **\$var | .foo = 1** instead.

If the right-hand side of '=' produces multiple values, then for each such value jq will set the paths on the left-hand side to the value and then it will output the modified **..**. For example, **(.a,.b)=range(2)** outputs **{"a":0,"b":0}**, then **{"a":1,"b":1}**. The "update" assignment forms (see below) do not do this.

Note too that **.a,.b=0** does not set **.a** and **.b**, but **(.a,.b)=0** sets both.

|=

As well as the assignment operator '=', jq provides the "update" operator '|=', which takes a filter on the right-hand side and works out the new value for the property of **.** being assigned to by running the old value through this expression. For instance, **.foo |= .+1** will build an object with the "foo" field set to the input's "foo" plus 1.

This example should show the difference between '=' and '|=':

Provide input **'{"a": {"b": 10}, "b": 20}'** to the programs:

```
.a = .b .a |= .b
```

The former will set the "a" field of the input to the "b" field of the input, and produce the output **{"a": 20}**. The latter will set the "a" field of the input to the "a" field's "b" field, producing **{"a": 10}**.

The left-hand side can be any general path expression; see **path()**.

Note that the left-hand side of '|=' refers to a value in **..**. Thus **\$var.foo |= . + 1** won't work as expected (**\$var.foo** is not a valid or useful path expression in **..**); use **\$var | .foo |= . + 1** instead.

If the right-hand side outputs multiple values, only the last one will be used.

```
jq '(.|select(type=="boolean")) |= if . then 1 else 0 end'
[true,false,[5,true,[true,[false]],false]]
=> [1,0,[5,1,[1,[0]],0]]
```

`+=, -=, *=, /=, %=, //=`

jq has a few operators of the form **a op= b**, which are all equivalent to **a |= . op b**. So, `+= 1` can be used to increment values.

```
jq 'foo += 1'
{"foo": 42}
=> {"foo": 43}
```

Complex assignments

Lots more things are allowed on the left-hand side of a jq assignment than in most languages. We've already seen simple field accesses on the left hand side, and it's no surprise that array accesses work just as well:

```
.posts[0].title = "JQ Manual"
```

What may come as a surprise is that the expression on the left may produce multiple results, referring to different points in the input document:

```
.posts[] | .comments |= . + ["this is great"]
```

That example appends the string "this is great" to the "comments" array of each post in the input (where the input is an object with a field "posts" which is an array of posts).

When jq encounters an assignment like `a = b`, it records the "path" taken to select a part of the input document while executing a. This path is then used to find which part of the input to change while executing the assignment. Any filter may be used on the left-hand side of an equals – whichever paths it selects from the input will be where the assignment is performed.

This is a very powerful operation. Suppose we wanted to add a comment to blog posts, using the same "blog" input above. This time, we only want to comment on the posts written by "stedolan". We can find those posts using the "select" function described earlier:

```
.posts[] | select(.author == "stedolan")
```

The paths provided by this operation point to each of the posts that "stedolan" wrote, and we can comment on each of them in the same way that we did before:

```
(.posts[] | select(.author == "stedolan") | .comments) |=
. + ["terrible."]
```

MODULES

jq has a library/module system. Modules are files whose names end in **.jq**.

Modules imported by a program are searched for in a default search path (see below). The **import** and

include directives allow the importer to alter this path.

Paths in the a search path are subject to various substitutions.

For paths starting with "~/", the user's home directory is substituted for "~".

For paths starting with "\$ORIGIN/", the path of the jq executable is substituted for "\$ORIGIN".

For paths starting with "./" or paths that are ".", the path of the including file is substituted for ".". For top-level programs given on the command-line, the current directory is used.

Import directives can optionally specify a search path to which the default is appended.

The default search path is the search path given to the **-L** command-line option, else ["~/jq", "\$ORIGIN/../lib/jq", "\$ORIGIN/../lib"].

Null and empty string path elements terminate search path processing.

A dependency with relative path "foo/bar" would be searched for in "foo/bar.jq" and "foo/bar/bar.jq" in the given search path. This is intended to allow modules to be placed in a directory along with, for example, version control files, README files, and so on, but also to allow for single-file modules.

Consecutive components with the same name are not allowed to avoid ambiguities (e.g., "foo/foo").

For example, with **-L\$HOME/.jq** a module **foo** can be found in **\$HOME/.jq/foo.jq** and **\$HOME/.jq/foo/foo.jq**.

If "\$HOME/.jq" is a file, it is sourced into the main program.

import RelativePathString as NAME [<metadata>];

Imports a module found at the given path relative to a directory in a search path. A ".jq" suffix will be added to the relative path string. The module's symbols are prefixed with "NAME::".

The optional metadata must be a constant jq expression. It should be an object with keys like "homepage" and so on. At this time jq only uses the "search" key/value of the metadata. The metadata is also made available to users via the **modulemeta** builtin.

The "search" key in the metadata, if present, should have a string or array value (array of strings); this is the search path to be prefixed to the top-level search path.

include RelativePathString [<metadata>];

Imports a module found at the given path relative to a directory in a search path as if it were included in place. A ".jq" suffix will be added to the relative path string. The module's symbols are imported into the caller's namespace as if the module's content had been included directly.

The optional metadata must be a constant jq expression. It should be an object with keys like "homepage" and so on. At this time jq only uses the "search" key/value of the metadata. The metadata is also made available to users via the **modulemeta** builtin.

import RelativePathString as \$NAME [<metadata>];

Imports a JSON file found at the given path relative to a directory in a search path. A ".json" suffix will be added to the relative path string. The file's data will be available as **\$NAME::NAME**.

The optional metadata must be a constant jq expression. It should be an object with keys like "homepage" and so on. At this time jq only uses the "search" key/value of the metadata. The metadata is also made available to users via the **modulemeta** builtin.

The "search" key in the metadata, if present, should have a string or array value (array of strings); this is the search path to be prefixed to the top-level search path.

module <metadata>;

This directive is entirely optional. It's not required for proper operation. It serves only the purpose of providing metadata that can be read with the **modulemeta** builtin.

The metadata must be a constant jq expression. It should be an object with keys like "homepage". At this time jq doesn't use this metadata, but it is made available to users via the **modulemeta** builtin.

modulemeta

Takes a module name as input and outputs the module's metadata as an object, with the module's imports (including metadata) as an array value for the "deps" key.

Programs can use this to query a module's metadata, which they could then use to, for example, search for, download, and install missing dependencies.

BUGS

Presumably. Report them or discuss them at:

<https://github.com/stedolan/jq/issues>

AUTHOR

Stephen Dolan <mu@netsoc.tcd.ie>

NAME

kalc - Programmable scientific RPN calculator

SYNOPSIS

kalc [*options*] [*commands*]

or

kalc --help | --version

DESCRIPTION

kalc is a programmable scientific calculator, using *RPN (Reverse Polish Notation)*. It includes over 200 functions and a built-in help system. **kalc** works with real numbers, complex numbers and also integers in other numeric bases. It has a complete programming language, with control-flow structures such as *if* and *while*. It has "unlimited" memory for you to store objects.

The behavior of **kalc** has been made to imitate as much as possible a HP48/HP49 calculator. This program is not, however, a HP emulator, and does not use any code from that calculator's ROM.

OPTIONS

- d <n> Display at most <n> stack levels. Specify -1 as this argument to have the whole stack shown.
- s Do not load the stack from the init file.
- m Do not load the memory from the init file.
- b Runs **kalc** in batch mode. After running the commands given in the command line, exits the program.
- help Displays a brief help message and exits successfully.
- version Displays version information and exits successfully.

AUTHOR

Eduardo M Kalinowski <ekalin@iname.com>

SEE ALSO

bc(1), **kalc.1.pdf**, <https://wa2l-edrc.sourceforge.net/manuals/man1/kalc.1.pdf>

The **kalc User's Manual (kalc.1.pdf)** which should come with this package describes in detail usage of **kalc** and all its functions.

NAME

ksh, rksh, pfksh – KornShell, a standard/restricted command and programming language

SYNOPSIS

[**±abcefhikmnoprstuvxBCDP**] [**-R** file] [**±o** option] ... [**-**] [arg ...]

rksh [**±abcefhikmnoprstuvxBCD**] [**-R** file] [**±o** option] ... [**-**] [arg ...]

DESCRIPTION

Ksh is a command and programming language that executes commands read from a terminal or a file. *Rksh* is a restricted version of the command interpreter *ksh*; it is used to set up login names and execution environments whose capabilities are more controlled than those of the standard shell. *Rpfksh* is a profile shell version of the command interpreter *ksh*; it is used to execute commands with the attributes specified by the user's profiles (see *pfexec*(1)). See *Invocation* below for the meaning of arguments to the shell.

Definitions.

A *metacharacter* is one of the following characters:

; & () | < > new-line space tab

A *blank* is a **tab** or a **space**. An *identifier* is a sequence of letters, digits, or underscores starting with a letter or underscore. Identifiers are used as components of *variable* names. A *vname* is a sequence of one or more identifiers separated by a **.** and optionally preceded by a **..**. Vnames are used as function and variable names. A *word* is a sequence of *characters* from the character set defined by the current locale, excluding non-quoted *metacharacters*.

A *command* is a sequence of characters in the syntax of the shell language. The shell reads each command and carries out the desired action either directly or by invoking separate utilities. A built-in command is a command that is carried out by the shell itself without creating a separate process. Some commands are built-in purely for convenience and are not documented here. Built-ins that cause side effects in the shell environment and built-ins that are found before performing a path search (see *Execution* below) are documented here. For historical reasons, some of these built-ins behave differently than other built-ins and are called *special built-ins*.

Commands.

A *simple-command* is a list of variable assignments (see *Variable Assignments* below) or a sequence of *blank* separated words which may be preceded by a list of variable assignments (see *Environment* below). The first word specifies the name of the command to be executed. Except as specified below, the remaining words are passed as arguments to the invoked command. The command name is passed as argument 0 (see *exec*(2)). The *value* of a simple-command is its exit status; 0-255 if it terminates normally; 256+*signum* if it terminates abnormally (the name of the signal corresponding to the exit status can be obtained via the **-l** option of the **kill** built-in utility).

A *pipeline* is a sequence of one or more *commands* separated by **|**. The standard output of each command but the last is connected by a *pipe*(2) to the standard input of the next command. Each command, except possibly the last, is run as a separate process; the shell waits for the last command to terminate. The exit status of a pipeline is the exit status of the last command unless the **pipefail** option is enabled. Each pipeline can be preceded by the *reserved word* **!** which causes the exit status of the pipeline to become 0 if the exit status of the last command is non-zero, and 1 if the exit status of the last command is 0.

A *list* is a sequence of one or more pipelines separated by **;**, **&**, **|&**, **&&**, or **| |**, and optionally terminated by **;**, **&**, or **|&**. Of these five symbols, **;**, **&**, and **|&** have equal precedence, which is lower than that of **&&** and **| |**. The symbols **&&** and **| |** also have equal precedence. A semicolon (**;**) causes sequential execution of the preceding pipeline; an ampersand (**&**) causes asynchronous execution of the preceding pipeline (i.e., the shell does *not* wait for that pipeline to finish). The symbol **|&** causes asynchronous execution of the preceding pipeline with a two-way pipe established to the parent shell; the standard input and output of the spawned pipeline can be written to and read from by the parent shell by applying the redirection operators **<&** and **>&** with arg **p** to commands and by using **-p** option of the built-in commands **read** and **print** described later. The symbol **&&** (**| |**) causes the *list* following it to be executed only if the preceding pipeline returns a zero (non-zero) value. One or more new-lines may appear in a *list* instead of a semicolon, to delimit a command. The first *item* of the first *pipeline* of a *list* that is a simple command not

beginning with a redirection, and not occurring within a **while**, **until**, or **if** *list*, can be predated by a semicolon. This semicolon is ignored unless the **showme** option is enabled as described with the **set** built-in below.

A *command* is either a simple-command or one of the following. Unless otherwise stated, the value returned by a command is that of the last simple-command executed in the command.

for *vname* [**in** *word* ...] **;do** *list* **;done**

Each time a **for** command is executed, *vname* is set to the next *word* taken from the **in** *word* list. If **in** *word* ... is omitted, then the **for** command executes the **do** *list* once for each positional parameter that is set starting from **1** (see *Parameter Expansion* below). Execution ends when there are no more words in the list.

for (([*expr1*] ; [*expr2*] ; [*expr3*])) **;do** *list* **;done**

The arithmetic expression *expr1* is evaluated first (see *Arithmetic evaluation* below). The arithmetic expression *expr2* is repeatedly evaluated until it evaluates to zero and when non-zero, *list* is executed and the arithmetic expression *expr3* evaluated. If any expression is omitted, then it behaves as if it evaluated to 1.

select *vname* [**in** *word* ...] **;do** *list* **;done**

A **select** command prints on standard error (file descriptor 2) the set of *words*, each preceded by a number. If **in** *word* ... is omitted, then the positional parameters starting from **1** are used instead (see *Parameter Expansion* below). The **PS3** prompt is printed and a line is read from the standard input. If this line consists of the number of one of the listed *words*, then the value of the variable *vname* is set to the *word* corresponding to this number. If this line is empty, the selection list is printed again. Otherwise the value of the variable *vname* is set to *null*. The contents of the line read from standard input is saved in the variable **REPLY**. The *list* is executed for each selection until a **break** or *end-of-file* is encountered. If the **REPLY** variable is set to *null* by the execution of *list*, then the selection list is printed before displaying the **PS3** prompt for the next selection.

case *word* **in** [([*pattern* [| *pattern*] ...) *list* ;;] ... **esac**

A **case** command executes the *list* associated with the first *pattern* that matches *word*. The form of the patterns is the same as that used for file-name generation (see *File Name Generation* below). The **;;** operator causes execution of **case** to terminate. If **&** is used in place of **;;** the next subsequent list, if any, is executed.

if *list* **;then** *list* [**elif** *list* **;then** *list*] ... [**else** *list*] **fi**

The *list* following **if** is executed and, if it returns a zero exit status, the *list* following the first **then** is executed. Otherwise, the *list* following **elif** is executed and, if its value is zero, the *list* following the next **then** is executed. Failing each successive **elif** *list*, the **else** *list* is executed. If the *list* has non-zero exit status and there is no **else** *list*, then the **if** command returns a zero exit status.

while *list* **;do** *list* **;done**

until *list* **;do** *list* **;done**

A **while** command repeatedly executes the **while** *list* and, if the exit status of the last command in the *list* is zero, executes the **do** *list*; otherwise the loop terminates. If no commands in the **do** *list* are executed, then the **while** command returns a zero exit status; **until** may be used in place of **while** to negate the loop termination test.

((*expression*))

The *expression* is evaluated using the rules for arithmetic evaluation described below. If the value of the arithmetic expression is non-zero, the exit status is 0, otherwise the exit status is 1.

(*list*)

Execute *list* in a separate environment. Note, that if two adjacent open parentheses are needed for nesting, a space must be inserted to avoid evaluation as an arithmetic command as described above.

{ *list* ; }

list is simply executed. Note that unlike the metacharacters (and), { and } are *reserved words* and

must occur at the beginning of a line or after a `;` in order to be recognized.

[[*expression*]]

Evaluates *expression* and returns a zero exit status when *expression* is true. See *Conditional Expressions* below, for a description of *expression*.

function *varname* { *list* ;}

varname () { *list* ;}

Define a function which is referenced by *varname*. A function whose *varname* contains a `.` is called a discipline function and the portion of the *varname* preceding the last `.` must refer to an existing variable. The body of the function is the *list* of commands between `{` and `}`. A function defined with the **function** *varname* syntax can also be used as an argument to the `.` special built-in command to get the equivalent behavior as if the *varname*() syntax were used to define it. (See *Functions* below.)

time [*pipeline*]

If *pipeline* is omitted the user and system time for the current shell and completed child processes is printed on standard error. Otherwise, *pipeline* is executed and the elapsed time as well as the user and system time are printed on standard error. The **TIMEFORMAT** variable may be set to a format string that specifies how the timing information should be displayed. See **Shell Variables** below for a description of the **TIMEFORMAT** variable.

The following reserved words are recognized as reserved only when they are the first word of a command and are not quoted:

if then else elif fi case esac for while until do done { } function select time
[[]] !

Variable Assignments.

One or more variable assignments can start a simple command or can be arguments to the **typeset**, **export**, or **readonly** special built-in commands. The syntax for an *assignment* is of the form:

varname=*word*

varname[*word*]=*word*

No space is permitted between *varname* and the `=` or between `=` and *word*.

varname=(*assign_list*)

No space is permitted between *varname* and the `=`. An *assign_list* can be one of the following:

word ...

Indexed array assignment.

[*word*]=*word* ...

Associative array assignment. If preceded by **typeset -a** this will create an indexed array instead.

assignment ...

Compound variable assignment. This creates a compound variable *varname* with sub-variables of the form *varname.name*, where *name* is the name portion of *assignment*. The value of *varname* will contain all the assignment elements. Additional assignments made to sub-variables of *varname* will also be displayed as part of the value of *varname*. If no *assignments* are specified, *varname* will be a compound variable allowing subsequence child elements to be defined.

typeset [*options*] *assignment* ...

Nested variable assignment. Multiple assignments can be specified by separating each of them with a `;`. The previous value is unset before the assignment.

In addition, a `+=` can be used in place of the `=` to signify adding to or appending to the previous value. When `+=` is applied to an arithmetic type, *word* is evaluated as an arithmetic expression and added to the current value. When applied to a string variable, the value defined by *word* is appended to the value. For compound assignments, the previous value is not unset and the new values are appended to the current ones provided that the types are compatible.

Comments.

A word beginning with # causes that word and all the following characters up to a new-line to be ignored.

Aliasing.

The first word of each command is replaced by the text of an **alias** if an **alias** for this word has been defined. An **alias** name consists of any number of characters excluding metacharacters, quoting characters, file expansion characters, parameter expansion and command substitution characters, and =. The replacement string can contain any valid shell script including the metacharacters listed above. The first word of each command in the replaced text, other than any that are in the process of being replaced, will be tested for aliases. If the last character of the alias value is a *blank* then the word following the alias will also be checked for alias substitution. Aliases can be used to redefine built-in commands but cannot be used to redefine the reserved words listed above. Aliases can be created and listed with the **alias** command and can be removed with the **unalias** command.

Aliasing is performed when scripts are read, not while they are executed. Therefore, for an alias to take effect, the **alias** definition command has to be executed before the command which references the alias is read.

The following aliases are compiled into the shell but can be unset or redefined:

```
autoload='typeset -fu'
command='command '
fc=hist
float='typeset -IE'
functions='typeset -f'
hash='alias -t --'
history='hist -l'
integer='typeset -li'
nameref='typeset -n'
nohup='nohup '
r='hist -s'
redirect='command exec'
source='command .'
stop='kill -s STOP'
suspend='kill -s STOP $$'
times='{ { time;} 2>&1;}'
type='whence -v'
```

Tilde Substitution.

After alias substitution is performed, each word is checked to see if it begins with an unquoted ~. For tilde substitution, *word* also refers to the *word* portion of parameter expansion (see *Parameter Expansion* below). If it does, then the word up to a / is checked to see if it matches a user name in the password database (See *getpwnam(3)*.) If a match is found, the ~ and the matched login name are replaced by the login directory of the matched user. If no match is found, the original text is left unchanged. A ~ by itself, or in front of a /, is replaced by \$HOME. A ~ followed by a + or - is replaced by the value of \$PWD and \$OLDPWD respectively.

In addition, when expanding a *variable assignment*, *tilde* substitution is attempted when the value of the assignment begins with a ~, and when a ~ appears after a :. The : also terminates a ~ login name.

Command Substitution.

The standard output from a command enclosed in parentheses preceded by a dollar sign (\$()) or a pair of grave accents (` `) may be used as part or all of a word; trailing new-lines are removed. In the second (obsolete) form, the string between the quotes is processed for special quoting characters before the command is executed (see *Quoting* below). The command substitution \$(*cat file*) can be replaced by the equivalent but faster \$(*<file*). The command substitution \$(*n<#*) will expand to the current byte offset for file descriptor *n*.

Arithmetic Substitution.

An arithmetic expression enclosed in double parentheses preceded by a dollar sign (\$(())) is replaced by the value of the arithmetic expression within the double parentheses.

Process Substitution.

This feature is only available on versions of the UNIX operating system that support the **/dev/fd** directory for naming open files. Each command argument of the form **<(list)** or **>(list)** will run process *list* asynchronously connected to some file in **/dev/fd**. The name of this file will become the argument to the command. If the form with **>** is selected then writing on this file will provide input for *list*. If **<** is used, then the file passed as an argument will contain the output of the *list* process. For example,

```
paste <(cut -f1 file1) <(cut -f3 file2) | tee >(process1) >(process2)
```

cuts fields 1 and 3 from the files *file1* and *file2* respectively, *pastes* the results together, and sends it to the processes *process1* and *process2*, as well as putting it onto the standard output. Note that the file, which is passed as an argument to the command, is a UNIX *pipe*(2) so programs that expect to *lseek*(2) on the file will not work.

Parameter Expansion.

A *parameter* is a *variable*, one or more digits, or any of the characters *****, **@**, **#**, **?**, **-**, **\$**, and **!**. A *variable* is denoted by a *vname*. To create a variable whose *vname* contains a **.**, a variable whose *vname* consists of everything before the last **.** must already exist. A *variable* has a *value* and zero or more *attributes*. *Variables* can be assigned *values* and *attributes* by using the **typeset** special built-in command. The attributes supported by the shell are described later with the **typeset** special built-in command. Exported variables pass values and attributes to the environment.

The shell supports both indexed and associative arrays. An element of an array variable is referenced by a *subscript*. A *subscript* for an indexed array is denoted by an *arithmetic expression* (see *Arithmetic evaluation* below) between a **[** and a **]**. To assign values to an indexed array, use **set -A vname value ...**. The value of all subscripts must be in the range of 0 through 1,048,575. Indexed arrays need not be declared. Any reference to a variable with a valid subscript is legal and an array will be created if necessary.

An associative array is created with the **-A** option to **typeset**. A *subscript* for an associative array is denoted by a string enclosed between **[** and **]**.

Referencing any array without a subscript is equivalent to referencing the array with subscript 0.

The *value* of a *variable* may be assigned by writing:

```
vname=value [ vname=value ] ...
```

or

```
vname[subscript]=value [ vname[subscript]=value ] ...
```

Note that no space is allowed before or after the **=**.

A *nameref* is a variable that is a reference to another variable. A *nameref* is created with the **-n** attribute of **typeset**. The value of the variable at the time of the **typeset** command becomes the variable that will be referenced whenever the *nameref* variable is used. The name of a *nameref* cannot contain a **.**. When a variable or function name contains a **.**, and the portion of the name up to the first **.** matches the name of a *nameref*, the variable referred to is obtained by replacing the *nameref* portion with the name of the variable referenced by the *nameref*. If a *nameref* is used as the index of a **for** loop, a name reference is established for each item in the list. A *nameref* provides a convenient way to refer to the variable inside a function whose name is passed as an argument to a function. For example, if the name of a variable is passed as the first argument to a function, the command

```
typeset -n var=$1
```

inside the function causes references and assignments to **var** to be references and assignments to the variable whose name has been passed to the function.

If either of the floating point attributes, **-E**, or **-F**, or the integer attribute, **-i**, is set for *vname*, then the *value* is subject to arithmetic evaluation as described below.

Positional parameters, parameters denoted by a number, may be assigned values with the **set** special built-in command. Parameter **\$0** is set from argument zero when the shell is invoked.

The character **\$** is used to introduce substitutable *parameters*.

\${parameter}

The shell reads all the characters from **\${** to the matching **}** as part of the same word even if it contains braces or metacharacters. The value, if any, of the parameter is substituted. The braces are required when *parameter* is followed by a letter, digit, or underscore that is not to be interpreted as part of its name, when the variable name contains a **.**. The braces are also required when a variable is subscripted unless it is part of an Arithmetic Expression or a Conditional Expression. If *parameter* is one or more digits then it is a positional parameter. A positional parameter of more than one digit must be enclosed in braces. If *parameter* is ***** or **@**, then all the positional parameters, starting with **\$1**, are substituted (separated by a field separator character). If an array *vname* with subscript ***** or **@** is used, then the value for each of the elements is substituted, separated by the first character of the value of **IFS**.

\${#parameter}

If *parameter* is ***** or **@**, the number of positional parameters is substituted. Otherwise, the length of the value of the *parameter* is substituted.

\${#vname[*]}**\${#vname[@]}**

The number of elements in the array *vname* is substituted.

\${!vname}

Expands to the name of the variable referred to by *vname*. This will be *vname* except when *vname* is a name reference.

\${!vname[subscript]}

Expands to name of the subscript unless *subscript* is ***** or **@**. When *subscript* is *****, the list of array subscripts for *vname* is generated. For a variable that is not an array, the value is 0 if the variable is set. Otherwise it is null. When *subscript* is **@**, same as above, except that when used in double quotes, each array subscript yields a separate argument.

\${!prefix*}

Expands to the names of the variables whose names begin with *prefix*.

\${parameter:-word}

If *parameter* is set and is non-null then substitute its value; otherwise substitute *word*.

\${parameter:=word}

If *parameter* is not set or is null then set it to *word*; the value of the parameter is then substituted. Positional parameters may not be assigned to in this way.

\${parameter:?word}

If *parameter* is set and is non-null then substitute its value; otherwise, print *word* and exit from the shell (if not interactive). If *word* is omitted then a standard message is printed.

\${parameter:+word}

If *parameter* is set and is non-null then substitute *word*; otherwise substitute nothing.

In the above, *word* is not evaluated unless it is to be used as the substituted string, so that, in the following example, **pwd** is executed only if **d** is not set or is null:

```
print ${d:-$(pwd)}
```

If the colon (**:**) is omitted from the above expressions, then the shell only checks whether *parameter* is set or not.

\${parameter:offset:length}**\${parameter:offset}**

Expands to the portion of the value of *parameter* starting at the character (counting from **0**) determined by expanding *offset* as an arithmetic expression and consisting of the number of characters determined by the arithmetic expression defined by *length*. In the second form, the remainder of the value is used. If A negative *offset* counts backwards from the end of *parameter*. Note that one or more *blanks* is required in front of a minus sign to prevent the shell from interpreting the operator as **:-**. If *parameter* is ***** or **@**, or is an array name indexed by ***** or **@**, then *offset* and *length*

refer to the array index and number of elements respectively. A negative *offset* is taken relative to one greater than the highest subscript for indexed arrays. The order for associate arrays is unspecified.

\${parameter#pattern}

\${parameter##pattern}

If the shell *pattern* matches the beginning of the value of *parameter*, then the value of this expansion is the value of the *parameter* with the matched portion deleted; otherwise the value of this *parameter* is substituted. In the first form the smallest matching pattern is deleted and in the second form the largest matching pattern is deleted. When *parameter* is @, *, or an array variable with subscript @ or *, the substring operation is applied to each element in turn.

\${parameter%pattern}

\${parameter%%pattern}

If the shell *pattern* matches the end of the value of *parameter*, then the value of this expansion is the value of the *parameter* with the matched part deleted; otherwise substitute the value of *parameter*. In the first form the smallest matching pattern is deleted and in the second form the largest matching pattern is deleted. When *parameter* is @, *, or an array variable with subscript @ or *, the substring operation is applied to each element in turn.

\${parameter/pattern/string}

\${parameter//pattern/string}

\${parameter/#pattern/string}

\${parameter/%pattern/string}

Expands *parameter* and replaces the longest match of *pattern* with the given *string*. Each occurrence of *pattern* in *string* is replaced by the portion of *parameter* that matches the *n*-th sub-pattern. In the first form, only the first occurrence of *pattern* is replaced. In the second form, each match for *pattern* is replaced by the given *string*. The third form restricts the pattern match to the beginning of the string while the fourth form restricts the pattern match to the end of the string. When *string* is null, the *pattern* will be deleted and the / in front of *string* may be omitted. When *parameter* is @, *, or an array variable with subscript @ or *, the substitution operation is applied to each element in turn. In this case, the *string* portion of *word* will be re-evaluated for each element.

The following parameters are automatically set by the shell:

- #** The number of positional parameters in decimal.
- Options supplied to the shell on invocation or by the **set** command.
- ?** The decimal value returned by the last executed command.
- \$** The process number of this shell.
- _** Initially, the value of **_** is an absolute pathname of the shell or script being executed as passed in the *environment*. Subsequently it is assigned the last argument of the previous command. This parameter is not set for commands which are asynchronous. This parameter is also used to hold the name of the matching **MAIL** file when checking for mail.
- !** The process number of the last background command invoked or the most recent job put in the background with the **bg** built-in command.

.sh.command

When processing a **DEBUG** trap, this variable contains the current command line that is about to run.

.sh.edchar

This variable contains the value of the keyboard character (or sequence of characters if the first character is an ESC, ascii **033**) that has been entered when processing a **KEYBD** trap (see *Key Bindings* below). If the value is changed as part of the trap action, then the new value replaces the key (or key sequence) that caused the trap.

.sh.edcol

The character position of the cursor at the time of the most recent **KEYBD** trap.

.sh.edmode

The value is set to ESC when processing a **KEYBD** trap while in **vi** insert mode. (See *Vi Editing Mode* below.) Otherwise, **.sh.edmode** is null when processing a **KEYBD** trap.

.sh.edtext

The characters in the input buffer at the time of the most recent **KEYBD** trap. The value is null when not processing a **KEYBD** trap.

.sh.file The pathname of the file that contains the current command.

.sh.fun The name of the current function that is being executed.

.sh.match

An indexed array which stores the most recent match and sub-pattern matches after conditional pattern matches that match and after variables expansions using the operators **#**, **%**, or **/**. The **0**-th element stores the complete match and the *i*-th element stores the *i*-th submatch. The **.sh.match** variable becomes unset when the variable that has expanded is assigned a new value.

.sh.name

Set to the name of the variable at the time that a discipline function is invoked.

.sh.subscript

Set to the name subscript of the variable at the time that a discipline function is invoked.

.sh.subshell

The current depth for subshells and command substitution.

.sh.value

Set to the value of the variable at the time that the **set** or **append** discipline function is invoked.

.sh.version

Set to a value that identifies the version of this shell.

LINENO

The current line number within the script or function being executed.

OLDPWD

The previous working directory set by the **cd** command.

OPTARG

The value of the last option argument processed by the **getopts** built-in command.

OPTIND

The index of the last option argument processed by the **getopts** built-in command.

PPID The process number of the parent of the shell.

PWD The present working directory set by the **cd** command.

RANDOM

Each time this variable is referenced, a random integer, uniformly distributed between 0 and 32767, is generated. The sequence of random numbers can be initialized by assigning a numeric value to **RANDOM**.

REPLY This variable is set by the **select** statement and by the **read** built-in command when no arguments are supplied.

SECONDS

Each time this variable is referenced, the number of seconds since shell invocation is returned. If this variable is assigned a value, then the value returned upon reference will be the value that was assigned plus the number of seconds since the assignment.

The following variables are used by the shell:

CDPATH

The search path for the **cd** command.

COLUMNS

If this variable is set, the value is used to define the width of the edit window for the shell edit modes and for printing **select** lists.

EDITOR

If the **VISUAL** variable is not set, the value of this variable will be checked for the patterns as described with **VISUAL** below and the corresponding editing option (see *Special*

- Command **set** below) will be turned on.
- ENV** If this variable is set, then parameter expansion, command substitution, and arithmetic substitution are performed on the value to generate the pathname of the script that will be executed when the shell is invoked (see *Invocation* below). This file is typically used for **alias** and **function** definitions. The default value is **\$HOME/.kshrc**.
- FCEDIT** Obsolete name for the default editor name for the **hist** command. **FCEDIT** is not used when **HISTEDIT** is set.
- FIGNORE** A pattern that defines the set of filenames that will be ignored when performing filename matching.
- FPATH** The search path for function definitions. The directories in this path are searched for a file with the same name as the function or command when a function with the **-u** attribute is referenced and when a command is not found. If an executable file with the name of that command is found, then it is read and executed in the current environment. Unlike **PATH**, the current directory must be represented explicitly by **.** rather than by adjacent **:** characters or a beginning or ending **:**.
- HISTCMD** Number of the current command in the history file.
- HISTEDIT** Name for the default editor name for the **hist** command.
- HISTFILE** If this variable is set when the shell is invoked, then the value is the pathname of the file that will be used to store the command history (see *Command Re-entry* below).
- HISTSIZE** If this variable is set when the shell is invoked, then the number of previously entered commands that are accessible by this shell will be greater than or equal to this number. The default is 512.
- HOME** The default argument (home directory) for the **cd** command.
- IFS** Internal field separators, normally **space**, **tab**, and **new-line** that are used to separate the results of command substitution or parameter expansion and to separate fields with the built-in command **read**. The first character of the **IFS** variable is used to separate arguments for the **"\$*"** substitution (see *Quoting* below). Each single occurrence of an **IFS** character in the string to be split, that is not in the *isspace* character class, and any adjacent characters in **IFS** that are in the *isspace* character class, delimit a field. One or more characters in **IFS** that belong to the *isspace* character class, delimit a field. In addition, if the same *isspace* character appears consecutively inside **IFS**, this character is treated as if it were not in the *isspace* class, so that if **IFS** consists of two **tab** characters, then two adjacent **tab** characters delimit a null field.
- LANG** This variable determines the locale category for any category not specifically selected with a variable starting with **LC_** or **LANG**.
- LC_ALL** This variable overrides the value of the **LANG** variable and any other **LC_** variable.
- LC_COLLATE** This variable determines the locale category for character collation information.
- LC_CTYPE** This variable determines the locale category for character handling functions. It determines the character classes for pattern matching (see *File Name Generation* below).
- LC_NUMERIC** This variable determines the locale category for the decimal point character.
- LINES** If this variable is set, the value is used to determine the column length for printing **select** lists. Select lists will print vertically until about two-thirds of **LINES** lines are filled.
- MAIL** If this variable is set to the name of a mail file *and* the **MAILPATH** variable is not set, then the shell informs the user of arrival of mail in the specified file.

MAILCHECK

This variable specifies how often (in seconds) the shell will check for changes in the modification time of any of the files specified by the **MAILPATH** or **MAIL** variables. The default value is 600 seconds. When the time has elapsed the shell will check before issuing the next prompt.

MAILPATH

A colon (:) separated list of file names. If this variable is set, then the shell informs the user of any modifications to the specified files that have occurred within the last **MAILCHECK** seconds. Each file name can be followed by a ? and a message that will be printed. The message will undergo parameter expansion, command substitution, and arithmetic substitution with the variable **\$_** defined as the name of the file that has changed. The default message is *you have mail in \$_*.

PATH The search path for commands (see *Execution* below). The user may not change **PATH** if executing under **rksh** (except in **.profile**).

PS1 The value of this variable is expanded for parameter expansion, command substitution, and arithmetic substitution to define the primary prompt string which by default is "\$ ". The character ! in the primary prompt string is replaced by the *command* number (see *Command Re-entry* below). Two successive occurrences of ! will produce a single ! when the prompt string is printed.

PS2 Secondary prompt string, by default "> ".

PS3 Selection prompt string used within a **select** loop, by default "#? ".

PS4 The value of this variable is expanded for parameter evaluation, command substitution, and arithmetic substitution and precedes each line of an execution trace. By default, **PS4** is "+ ". In addition when **PS4** is unset, the execution trace prompt is also "+ ".

SHELL The pathname of the *shell* is kept in the environment. At invocation, if the basename of this variable is **rsh**, **rksh**, or **krsh**, then the shell becomes restricted. If it is **pfsh** or **pfksh**, then the shell becomes a profile shell (see *pfexec(1)*).

TIMEFORMAT

The value of this parameter is used as a format string specifying how the timing information for pipelines prefixed with the **time** reserved word should be displayed. The % character introduces a format sequence that is expanded to a time value or other information. The format sequences and their meanings are as follows.

| | |
|----------|---|
| % % | A literal %. |
| %[p][l]R | The elapsed time in seconds. |
| %[p][l]U | The number of CPU seconds spent in user mode. |
| %[p][l]S | The number of CPU seconds spent in system mode. |
| %P | The CPU percentage, computed as (U + S) / R. |

The braces denote optional portions. The optional *p* is a digit specifying the *precision*, the number of fractional digits after a decimal point. A value of 0 causes no decimal point or fraction to be output. At most three places after the decimal point can be displayed; values of *p* greater than 3 are treated as 3. If *p* is not specified, the value 3 is used.

The optional **l** specifies a longer format, including hours if greater than zero, minutes, and seconds of the form *HHhMMmSS.FFs*. The value of *p* determines whether or not the fraction is included.

All other characters are output without change and a trailing newline is added. If unset, the default value, `$'\nreal\t%2lR\nuser\t%2lU\nsys\t%2lS'`, is used. If the value is null, no timing information is displayed.

TMOUT

If set to a value greater than zero, **TMOUT** will be the default timeout value for the **read** built-in command. The **select** compound command terminates after **TMOUT** seconds when input is from a terminal. Otherwise, the shell will terminate if a line is not entered

within the prescribed number of seconds while reading from a terminal. (Note that the shell can be compiled with a maximum bound for this value which cannot be exceeded.)

VISUAL

If the value of this variable matches the pattern **[Vv][li]**, then the **vi** option (see Special Command **set** below) is turned on. If the value matches the pattern **gmacs**, the **gmacs** option is turned on. If the value matches the pattern **macs**, then the **emacs** option will be turned on. The value of **VISUAL** overrides the value of **EDITOR**.

The shell gives default values to **PATH**, **PS1**, **PS2**, **PS3**, **PS4**, **MAILCHECK**, **FCEDIT**, **TMOU** and **IFS**, while **HOME**, **SHELL**, **ENV**, and **MAIL** are not set at all by the shell (although **HOME** is set by *login(1)*). On some systems **MAIL** and **SHELL** are also set by *login(1)*.

Field Splitting.

After parameter expansion and command substitution, the results of substitutions are scanned for the field separator characters (those found in **IFS**) and split into distinct fields where such characters are found. Explicit null fields ("" or "") are retained. Implicit null fields (those resulting from *parameters* that have no values or command substitutions with no output) are removed.

If the **braceexpand** (**-B**) option is set then each of the fields resulting from **IFS** are checked to see if they contain one or more of the brace patterns *{*,*}*, *{l1..l2}*, *{n1..n2}*, *{n1..n2%fmt}*, *{n1..n2 ..n3}*, or *{n1..n2 ..n3%fmt}*, where *** represents any character, *l1*, *l2* are letters and *n1*, *n2*, *n3* are signed numbers and *fmt* is a format specified as used by **printf**. In each case, fields are created by prepending the characters before the **{** and appending the characters after the **}** to each of the strings generated by the characters between the **{** and **}**. The resulting fields are checked to see if they have any brace patterns.

In the first form, a field is created for each string between **{** and **,**, between **,** and **,**, and between **,** and **}**. The string represented by *** can contain embedded matching **{** and **}** without quoting. Otherwise, each **{** and **}** with *** must be quoted.

In the seconds form, *l1* and *l2* must both be either upper case or both be lower case characters in the C locale. In this case a field is created for each character from *l1* thru *l2*.

In the remaining forms, a field is created for each number starting at *n1* and continuing until it reaches *n2* incrementing *n1* by *n3*. The cases where *n3* is not specified behave as if *n3* where **1** if *n1* ≤ *n2* and **-1** otherwise. If forms which specify *%fmt* any format flags, widths and precisions can be specified and *fmt* can end in any of the specifiers **cdiouxX**. For example, *{a,z}{1..5..3%02d}{b..c}x* expands to the 8 fields, **a01bx**, **a01cx**, **a04bx**, **a04cx**, **z01bx**, **z01cx**, **z04bx** and **z4cx**.

File Name Generation.

Following splitting, each field is scanned for the characters *****, **?**, **(**, and **[** unless the **-f** option has been set. If one of these characters appears, then the word is regarded as a *pattern*. Each file name component that contains any pattern character is replaced with a lexicographically sorted set of names that matches the pattern from that directory. If no file name is found that matches the pattern, then that component of the file name is left unchanged unless the pattern is prefixed with **~(N)** in which case it is removed as described below. If **IGNORE** is set, then each file name component that matches the pattern defined by the value of **IGNORE** is ignored when generating the matching filenames. The names **.** and **..** are also ignored. If **IGNORE** is not set, the character **.** at the start of each file name component will be ignored unless the first character of the pattern corresponding to this component is the character **.** itself. Note, that for other uses of pattern matching the **/** and **.** are not treated specially.

- *** Matches any string, including the null string. When used for filename expansion, if the **globstar** option is on, two adjacent *****'s by itself will match all files and zero or more directories and subdirectories. If followed by a **/** then only directories and subdirectories will match.
- ?** Matches any single character.
- [...]** Matches any one of the enclosed characters. A pair of characters separated by **-** matches any character lexically between the pair, inclusive. If the first character following the opening **[** is a **!** then any character not enclosed is matched. A **-** can be included in the character set by putting it as the first or last character.

Within [and], character classes can be specified with the syntax `[class]` where *class* is one of the following classes defined in the ANSI-C standard: (Note that **word** is equivalent to **alnum** plus the character `_`).

alnum alpha blank cntrl digit graph lower print punct space upper word xdigit

Within [and], an equivalence class can be specified with the syntax `[=c=]` which matches all characters with the same primary collation weight (as defined by the current locale) as the character *c*.

Within [and], `[.symbol.]` matches the collating symbol *symbol*.

A *pattern-list* is a list of one or more patterns separated from each other with a `&` or `|`. A `&` signifies that all patterns must be matched whereas `|` requires that only one pattern be matched. Composite patterns can be formed with one or more of the following sub-patterns:

`?(pattern-list)`

Optionally matches any one of the given patterns.

`*(pattern-list)`

Matches zero or more occurrences of the given patterns.

`+(pattern-list)`

Matches one or more occurrences of the given patterns.

`{n}(pattern-list)`

Matches *n* occurrences of the given patterns.

`{m,n}(pattern-list)`

Matches from *m* to *n* occurrences of the given patterns. If *m* is omitted, 0 will be used. If *n* is omitted at least *m* occurrences will be matched.

`@(pattern-list)`

Matches exactly one of the given patterns.

`!(pattern-list)`

Matches anything except one of the given patterns.

By default, each pattern, or sub-pattern will match the longest string possible consistent with generating the longest overall match. If more than one match is possible, the one starting closest to the beginning of the string will be chosen. However, for each of the above compound patterns a `-` can be inserted in front of the `(` to cause the shortest match to the specified *pattern-list* to be used.

When *pattern-list* is contained within parentheses, the backslash character `\` is treated specially even when inside a character class. All ANSI-C character escapes are recognized and match the specified character. In addition the following escape sequences are recognized:

| | |
|-----------------|--|
| <code>\d</code> | Matches any character in the digit class. |
| <code>\D</code> | Matches any character not in the digit class. |
| <code>\s</code> | Matches any character in the space class. |
| <code>\S</code> | Matches any character not in the space class. |
| <code>\w</code> | Matches any character in the word class. |
| <code>\W</code> | Matches any character not in the word class. |

A pattern of the form `%(pattern-pair(s))` is a sub-pattern that can be used to match nested character expressions. Each *pattern-pair* is a two character sequence which cannot contain `&` or `|`. The first *pattern-pair* specifies the starting and ending characters for the match. Each subsequent *pattern-pair* represents the beginning and ending characters of a nested group that will be skipped over when counting starting and ending character matches. The behavior is unspecified when the first character of a *pattern-pair* is alphanumeric except for the following:

| | |
|----------|---|
| D | Causes the ending character to terminate the search for this pattern without finding a match. |
| E | Causes the ending character to be interpreted as an escape character. |
| L | Causes the ending character to be interpreted as a quote character causing all characters to be ignored when looking for a match. |
| Q | Causes the ending character to be interpreted as a quote character causing all characters other than any escape character to be ignored when looking for a match. |

Thus, `%({}Q"E)`, matches characters starting at `{` until the matching `}` is found not counting any `{` or `}` that

is inside a double quoted string or preceded by the escape character `\`. Without the `{ }` this pattern matches any C language string.

Each sub-pattern in a composite pattern is numbered, starting at 1, by the location of the `(` within the pattern. The sequence `\n`, where `n` is a single digit and `\n` comes after the `n`-th. sub-pattern, matches the same string as the sub-pattern itself.

Finally a pattern can contain sub-patterns of the form `~(options:pattern-list)`, where either *options* or *pattern-list* can be omitted. Unlike the other compound patterns, these sub-patterns are not counted in the numbered sub-patterns. If *options* is present, it can consist of one or more of the following:

- +** Enable the following options. This is the default.
- Disable the following options.
- E** The remainder of the pattern uses extended regular expression syntax like the *egrep*(1) command.
- F** The remainder of the pattern uses *fgrep*(1) expression syntax.
- G** The remainder of the pattern uses basic regular expression syntax like the *grep*(1) command.
- K** The remainder of the pattern uses shell pattern syntax. This is the default.
- N** This is ignored. However, when it is the first letter and is used with file name generation, and no matches occur, the file pattern expands to the empty string.
- i** Treat the match as case insensitive.
- g** File the longest match (greedy). This is the default.
- l** Left anchor the pattern. This is the default for **K** style patterns.
- r** Right anchor the pattern. This is the default for **K** style patterns.

If both *options* and *pattern-list* are specified, then the options apply only to *pattern-list*. Otherwise, these options remain in effect until they are disabled by a subsequent `~(...)` or at the end of the sub-pattern containing `~(...)`.

Quoting.

Each of the *metacharacters* listed earlier (see *Definitions* above) has a special meaning to the shell

- i** Treat the match as case insensitive.
- g** File the longest match (greedy). This is the default.

If both *options* and *pattern-list* are specified, then the options apply only to *pattern-list*. Otherwise, these options remain in effect until they are disabled by a subsequent `~(...)` or at the end of the sub-pattern containing `~(...)`.

Quoting.

Each of the *metacharacters* listed earlier (see *Definitions* above) has a special meaning to the shell and causes termination of a word unless quoted. A character may be *quoted* (i.e., made to stand for itself) by preceding it with a `\`. The pair `\new-line` is removed. All characters enclosed between a pair of single quote marks (`'`) that is not preceded by a `$` are quoted. A single quote cannot appear within the single quotes. A single quoted string preceded by an unquoted `$` is processed as an ANSI-C string except for the following:

- \0** Causes the remainder of the string to be ignored.
- \E** Equivalent to the escape character (ascii **033**),
- \e** Equivalent to the escape character (ascii **033**),
- \cx** Expands to the character control-*x*.
- \C[.name.]** Expands to the collating element *name*.

Inside double quote marks (`"`), parameter and command substitution occur and `\` quotes the characters `\`, ```, `"`, and `$`. A `$` in front of a double quoted string will be ignored in the "C" or "POSIX" locale, and may cause the string to be replaced by a locale specific string otherwise. The meaning of `$*` and `$@` is identical when not quoted or when used as a variable assignment value or as a file name. However, when used as a command argument, `"$*"` is equivalent to `"$1d2d..."`, where *d* is the first character of the **IFS** variable, whereas `"$@"` is equivalent to `"$1" "$2" ...`. Inside grave quote marks (```), `\` quotes the characters `\`, ```, and `$`. If the grave quotes occur within double quotes, then `\` also quotes the character `"`.

The special meaning of reserved words or aliases can be removed by quoting any character of the reserved word. The recognition of function names or built-in command names listed below cannot be altered by quoting them.

Arithmetic Evaluation.

The shell performs arithmetic evaluation for arithmetic substitution, to evaluate an arithmetic command, to evaluate an indexed array subscript, and to evaluate arguments to the built-in commands **shift** and **let**. Evaluations are performed using double precision floating point arithmetic or long double precision floating point for systems that provide this data type. Floating point constants follow the ANSI-C programming language floating point conventions. Integer constants follow the ANSI-C programming language integer constant conventions although only single byte character constants are recognized and character casts are not recognized. In addition constants can be of the form `[base#]n` where *base* is a decimal number between two and sixty-four representing the arithmetic base and *n* is a number in that base. The digits above 9 are represented by the lower case letters, the upper case letters, `@`, and `_` respectively. For bases less than or equal to 36, upper and lower case characters can be used interchangeably.

An arithmetic expression uses the same syntax, precedence, and associativity of expression as the C language. All the C language operators that apply to floating point quantities can be used. In addition, the operator `**` can be used for exponentiation. It has higher precedence than multiplication and is left associative. In addition, when the value of an arithmetic variable or sub-expression can be represented as a long integer, all C language integer arithmetic operations can be performed. Variables can be referenced by name within an arithmetic expression without using the parameter expansion syntax. When a variable is referenced, its value is evaluated as an arithmetic expression.

Any of the following math library functions that are in the C math library can be used within an arithmetic expression:

```
abs acos acosh asin asinh atan atan2 atanh cbrt copysign cos cosh erf erfc
exp exp2 expm1 fabs fdim finite floor fma fmax fmod hypot ilogb int isinf
isnan lgamma log log2 logb nearbyint nextafter nexttoward pow rint round sin
sinh sqrt tan tanh tgamma trunc
```

An internal representation of a *variable* as a double precision floating point can be specified with the `-E [n]` or `-F [n]` option of the **typeset** special built-in command. The `-E` option causes the expansion of the value to be represented using scientific notation when it is expanded. The optional option argument *n* defines the number of significant figures. The `-F` option causes the expansion to be represented as a floating decimal number when it is expanded. The optional option argument *n* defines the number of places after the decimal point in this case.

An internal integer representation of a *variable* can be specified with the `-i [n]` option of the **typeset** special built-in command. The optional option argument *n* specifies an arithmetic base to be used when expanding the variable. If you do not specify an arithmetic base, base 10 will be used.

Arithmetic evaluation is performed on the value of each assignment to a variable with the `-E`, `-F`, or `-i` attribute. Assigning a floating point number to a variable whose type is an integer causes the fractional part to be truncated.

Prompting.

When used interactively, the shell prompts with the value of **PS1** after expanding it for parameter expansion, command substitution, and arithmetic substitution, before reading a command. In addition, each single `!` in the prompt is replaced by the command number. A `!!` is required to place `!` in the prompt. If at any time a new-line is typed and further input is needed to complete a command, then the secondary prompt (i.e., the value of **PS2**) is issued.

Conditional Expressions.

A *conditional expression* is used with the `[[` compound command to test attributes of files and to compare strings. Field splitting and file name generation are not performed on the words between `[[` and `]]`. Each expression can be constructed from one or more of the following unary or binary expressions:

string True, if *string* is not null.

-a file Same as **-e** below. This is obsolete.
-b file True, if *file* exists and is a block special file.
-c file True, if *file* exists and is a character special file.
-d file True, if *file* exists and is a directory.
-e file True, if *file* exists.
-f file True, if *file* exists and is an ordinary file.
-g file True, if *file* exists and it has its setgid bit set.
-k file True, if *file* exists and it has its sticky bit set.
-n string
 True, if length of *string* is non-zero.
-o ?option
 True, if option named *option* is a valid option name.
-o option
 True, if option named *option* is on.
-p file True, if *file* exists and is a fifo special file or a pipe.
-r file True, if *file* exists and is readable by current process.
-s file True, if *file* exists and has size greater than zero.
-t fildes
 True, if file descriptor number *fildes* is open and associated with a terminal device.
-u file True, if *file* exists and it has its setuid bit set.
-w file True, if *file* exists and is writable by current process.
-x file True, if *file* exists and is executable by current process. If *file* exists and is a directory, then true if the current process has permission to search in the directory.
-z string
 True, if length of *string* is zero.
-L file True, if *file* exists and is a symbolic link.
-h file True, if *file* exists and is a symbolic link.
-N file True, if *file* exists and the modification time is greater than the last access time.
-O file True, if *file* exists and is owned by the effective user id of this process.
-G file True, if *file* exists and its group matches the effective group id of this process.
-S file True, if *file* exists and is a socket.
file1 -nt file2
 True, if *file1* exists and *file2* does not, or *file1* is newer than *file2*.
file1 -ot file2
 True, if *file2* exists and *file1* does not, or *file1* is older than *file2*.
file1 -ef file2
 True, if *file1* and *file2* exist and refer to the same file.
string == pattern
 True, if *string* matches *pattern*. Any part of *pattern* can be quoted to cause it to be matched as a string. With a successful match to a pattern, the **.sh.match** array variable will contain the match and sub-pattern matches.
string = pattern
 Same as **==** above, but is obsolete.
string != pattern
 True, if *string* does not match *pattern*. When the *string* matches the *pattern* the **.sh.match** array variable will contain the match and sub-pattern matches.
string =~ ere
 True if *string* matches the pattern **~(E)ere** where *ere* is an extended regular expression.
string1 < string2
 True, if *string1* comes before *string2* based on ASCII value of their characters.
string1 > string2
 True, if *string1* comes after *string2* based on ASCII value of their characters.
 The following obsolete arithmetic comparisons are also permitted:

exp1 **-eq** *exp2*
 True, if *exp1* is equal to *exp2*.
exp1 **-ne** *exp2*
 True, if *exp1* is not equal to *exp2*.
exp1 **-lt** *exp2*
 True, if *exp1* is less than *exp2*.
exp1 **-gt** *exp2*
 True, if *exp1* is greater than *exp2*.
exp1 **-le** *exp2*
 True, if *exp1* is less than or equal to *exp2*.
exp1 **-ge** *exp2*
 True, if *exp1* is greater than or equal to *exp2*.

In each of the above expressions, if *file* is of the form **/dev/fd/*n***, where *n* is an integer, then the test is applied to the open file whose descriptor number is *n*.

A compound expression can be constructed from these primitives by using any of the following, listed in decreasing order of precedence.

(*expression*)
 True, if *expression* is true. Used to group expressions.
! *expression*
 True if *expression* is false.
expression1 **&&** *expression2*
 True, if *expression1* and *expression2* are both true.
expression1 **||** *expression2*
 True, if either *expression1* or *expression2* is true.

Input/Output.

Before a command is executed, its input and output may be redirected using a special notation interpreted by the shell. The following may appear anywhere in a simple-command or may precede or follow a *command* and are *not* passed on to the invoked command. Command substitution, parameter expansion, and arithmetic substitution occur before *word* or *digit* is used except as noted below. File name generation occurs only if the shell is interactive and the pattern matches a single file. Field splitting is not performed.

In each of the following redirections, if *file* is of the form **/dev/sctp/host/port**, **/dev/tcp/host/port**, or **/dev/udp/host/port**, where *host* is a hostname or host address, and *port* is a service given by name or an integer port number, then the redirection attempts to make a **tcp**, **sctp** or **udp** connection to the corresponding socket.

No intervening space is allowed between the characters of redirection operators.

| | |
|--|--|
| < <i>word</i> | Use file <i>word</i> as standard input (file descriptor 0). |
| > <i>word</i> | Use file <i>word</i> as standard output (file descriptor 1). If the file does not exist then it is created. If the file exists, and the noclobber option is on, this causes an error; otherwise, it is truncated to zero length. |
| > <i>word</i> | Sames as > , except that it overrides the noclobber option. |
| >> <i>word</i> | Use file <i>word</i> as standard output. If the file exists, then output is appended to it (by first seeking to the end-of-file); otherwise, the file is created. |
| <> <i>word</i> | Open file <i>word</i> for reading and writing as standard input. |
| << [-] <i>word</i> | The shell input is read up to a line that is the same as <i>word</i> after any quoting has been removed, or to an end-of-file. No parameter substitution, command substitution, arithmetic substitution or file name generation is performed on <i>word</i> . The resulting document, called a <i>here-document</i> , becomes the standard input. If any character of <i>word</i> is quoted, then no interpretation is placed upon the characters of the document; otherwise, parameter expansion, command substitution, and arithmetic substitution occur, \new-line is ignored, and \ must be used to quote the characters \ , \$, ` . If - is appended to << , then all leading |

tabs are stripped from *word* and from the document. If # is appended to <<, then leading spaces and tabs will be stripped off the first line of the document and up to an equivalent indentation will be stripped from the remaining lines and from *word*. A tab stop is assumed to occur at every 8 columns for the purposes of determining the indentation.

| | |
|----------------------|---|
| <<< <i>word</i> | A short form of here document in which <i>word</i> becomes the contents of the here-document after any parameter expansion, command substitution, and arithmetic substitution occur. |
| <& <i>digit</i> | The standard input is duplicated from file descriptor <i>digit</i> (see <i>dup(2)</i>). Similarly for the standard output using >& <i>digit</i> . |
| <& <i>digit</i> - | The file descriptor given by <i>digit</i> is moved to standard input. Similarly for the standard output using >& <i>digit</i> -. |
| <&- | The standard input is closed. Similarly for the standard output using >&-. |
| <&p | The input from the co-process is moved to standard input. |
| >&p | The output to the co-process is moved to standard output. |
| <# (<i>expr</i>) | Evaluate arithmetic expression <i>expr</i> and position file descriptor 0 to the resulting value bytes from the start of the file. The variables CUR and EOF evaluate to the current offset and end-of-file offset respectively when evaluating <i>expr</i> . |
| ># (<i>offset</i>) | The same as <# except applies to file descriptor 1. |
| <# <i>pattern</i> | Seeks forward to the beginning of the next line containing <i>pattern</i> . |
| <## <i>pattern</i> | The same as <# except that the portion of the file that is skipped is copied to standard output. |

If one of the above is preceded by a digit, with no intervening space, then the file descriptor number referred to is that specified by the digit (instead of the default 0 or 1). If one of the above, other than >&- and the ># and <# forms, is preceded by {*varname*} with no intervening space, then a file descriptor number > 10 will be selected by the shell and stored in the variable *varname*. If >&- or the any of the ># and <# forms is preceded by {*varname*} the value of *varname* defines the file descriptor to close or position. For example:

```
... 2>&1
```

means file descriptor 2 is to be opened for writing as a duplicate of file descriptor 1 and

```
exec {n}<file
```

means open file named **file** for reading and store the file descriptor number in variable **n**.

The order in which redirections are specified is significant. The shell evaluates each redirection in terms of the (*file descriptor*, *file*) association at the time of evaluation. For example:

```
... 1>fname 2>&1
```

first associates file descriptor 1 with file *fname*. It then associates file descriptor 2 with the file associated with file descriptor 1 (i.e. *fname*). If the order of redirections were reversed, file descriptor 2 would be associated with the terminal (assuming file descriptor 1 had been) and then file descriptor 1 would be associated with file *fname*.

If a command is followed by & and job control is not active, then the default standard input for the command is the empty file **/dev/null**. Otherwise, the environment for the execution of a command contains the file descriptors of the invoking shell as modified by input/output specifications.

Environment.

The *environment* (see *environ(7)*) is a list of name-value pairs that is passed to an executed program in the same way as a normal argument list. The names must be *identifiers* and the values are character strings. The shell interacts with the environment in several ways. On invocation, the shell scans the environment and creates a variable for each name found, giving it the corresponding value and attributes and marking it *export*. Executed commands inherit the environment. If the user modifies the values of these variables or

creates new ones, using the **export** or **typeset -x** commands, they become part of the environment. The environment seen by any executed command is thus composed of any name-value pairs originally inherited by the shell, whose values may be modified by the current shell, plus any additions which must be noted in **export** or **typeset -x** commands.

The environment for any *simple-command* or function may be augmented by prefixing it with one or more variable assignments. A variable assignment argument is a word of the form *identifier=value*. Thus:

```
TERM=450 cmd args          and
(export TERM; TERM=450; cmd args)
```

are equivalent (as far as the above execution of *cmd* is concerned except for special built-in commands listed below – those that are preceded with a dagger).

If the obsolete **-k** option is set, *all* variable assignment arguments are placed in the environment, even if they occur after the command name. The following first prints **a=b c** and then **c**:

```
echo a=b c
set -k
echo a=b c
```

This feature is intended for use with scripts written for early versions of the shell and its use in new scripts is strongly discouraged. It is likely to disappear someday.

Functions.

For historical reasons, there are two ways to define functions, the *name()* syntax and the **function name** syntax, described in the *Commands* section above. Shell functions are read in and stored internally. Alias names are resolved when the function is read. Functions are executed like commands with the arguments passed as positional parameters. (See *Execution* below.)

Functions defined by the **function name** syntax and called by name execute in the same process as the caller and share all files and present working directory with the caller. Traps caught by the caller are reset to their default action inside the function. A trap condition that is not caught or ignored by the function causes the function to terminate and the condition to be passed on to the caller. A trap on **EXIT** set inside a function is executed in the environment of the caller after the function completes. Ordinarily, variables are shared between the calling program and the function. However, the **typeset** special built-in command used within a function defines local variables whose scope includes the current function. They can be passed to functions that they call in the variable assignment list that precedes the call or as arguments passed as name references. Errors within functions return control to the caller.

Functions defined with the *name()* syntax and functions defined with the **function name** syntax that are invoked with the **.** special built-in are executed in the caller's environment and share all variables and traps with the caller. Errors within these function executions cause the script that contains them to abort.

The special built-in command **return** is used to return from function calls.

Function names can be listed with the **-f** or **+f** option of the **typeset** special built-in command. The text of functions, when available, will also be listed with **-f**. Functions can be undefined with the **-f** option of the **unset** special built-in command.

Ordinarily, functions are unset when the shell executes a shell script. Functions that need to be defined across separate invocations of the shell should be placed in a directory and the **FPATH** variable should contain the name of this directory. They may also be specified in the **ENV** file.

Discipline Functions.

Each variable can have zero or more discipline functions associated with it. The shell initially understands the discipline names **get**, **set**, **append**, and **unset** but on most systems others can be added at run time via the C programming interface extension provided by the **builtin** built-in utility. If the **get** discipline is defined for a variable, it is invoked whenever the given variable is referenced. If the variable **.sh.value** is assigned a value inside the discipline function, the referenced variable will evaluate to this value instead. If the **set** discipline is defined for a variable, it is invoked whenever the given variable is assigned a value. If the **append** discipline is defined for a variable, it is invoked whenever a value is appended to the given variable. The variable **.sh.value** is given the value of the variable before invoking the discipline, and the

variable will be assigned the value of **.sh.value** after the discipline completes. If **.sh.value** is unset inside the discipline, then that value is unchanged. If the **unset** discipline is defined for a variable, it is invoked whenever the given variable is unset. The variable will not be unset unless it is unset explicitly from within this discipline function.

The variable **.sh.name** contains the name of the variable for which the discipline function is called, **.sh.subscript** is the subscript of the variable, and **.sh.value** will contain the value being assigned inside the **set** discipline function. For the **set** discipline, changing **.sh.value** will change the value that gets assigned.

Jobs.

If the **monitor** option of the **set** command is turned on, an interactive shell associates a *job* with each pipeline. It keeps a table of current jobs, printed by the **jobs** command, and assigns them small integer numbers. When a job is started asynchronously with **&**, the shell prints a line which looks like:

```
[1] 1234
```

indicating that the job which was started asynchronously was job number 1 and had one (top-level) process, whose process id was 1234.

This paragraph and the next require features that are not in all versions of UNIX and may not apply. If you are running a job and wish to do something else you may hit the key **^Z** (control-Z) which sends a **STOP** signal to the current job. The shell will then normally indicate that the job has been 'Stopped', and print another prompt. You can then manipulate the state of this job, putting it in the background with the **bg** command, or run some other commands and then eventually bring the job back into the foreground with the foreground command **fg**. A **^Z** takes effect immediately and is like an interrupt in that pending output and unread input are discarded when it is typed.

A job being run in the background will stop if it tries to read from the terminal. Background jobs are normally allowed to produce output, but this can be disabled by giving the command **stty tostop**. If you set this **tty** option, then background jobs will stop when they try to produce output like they do when they try to read input.

There are several ways to refer to jobs in the shell. A job can be referred to by the process id of any process of the job or by one of the following:

%number

The job with the given number.

%string

Any job whose command line begins with *string*.

%?string

Any job whose command line contains *string*.

% %

Current job.

%+

Equivalent to **% %**.

%-

Previous job.

The shell learns immediately whenever a process changes state. It normally informs you whenever a job becomes blocked so that no further progress is possible, but only just before it prints a prompt. This is done so that it does not otherwise disturb your work. The **notify** option of the **set** command causes the shell to print these job change messages as soon as they occur.

When the **monitor** option is on, each background job that completes triggers any trap set for **CHLD**.

When you try to leave the shell while jobs are running or stopped, you will be warned that 'You have stopped(running) jobs.' You may use the **jobs** command to see what they are. If you immediately try to exit again, the shell will not warn you a second time, and the stopped jobs will be terminated. When a login shell receives a HUP signal, it sends a HUP signal to each job that has not been disowned with the **disown** built-in command described below.

Signals.

The **INT** and **QUIT** signals for an invoked command are ignored if the command is followed by **&** and the **monitor** option is not active. Otherwise, signals have the values inherited by the shell from its parent (but see also the **trap** built-in command below).

Execution.

Each time a command is read, the above substitutions are carried out. If the command name matches one of the *Special Built-in Commands* listed below, it is executed within the current shell process. Next, the command name is checked to see if it matches a user defined function. If it does, the positional parameters are saved and then reset to the arguments of the *function* call. A function is also executed in the current shell process. When the *function* completes or issues a **return**, the positional parameter list is restored. For functions defined with the **function name** syntax, any trap set on **EXIT** within the function is executed. The exit value of a *function* is the value of the last command executed. If a command name is not a *special built-in command* or a user defined *function*, but it is one of the built-in commands listed below, it is executed in the current shell process.

The shell variable **PATH** defines the search path for the directory containing the command. Alternative directory names are separated by a colon (:). The default path is **/bin:/usr/bin:** (specifying **/bin**, **/usr/bin**, and the current directory in that order). The current directory can be specified by two or more adjacent colons, or by a colon at the beginning or end of the path list. If the command name contains a **/**, then the search path is not used. Otherwise, each directory in the path is searched for an executable file of the given name that is not a directory. If found, and if the shell determines that there is a built-in version of a command corresponding to a given pathname, this built-in is invoked in the current process. If found, and this directory is also contained in the value of the **FPATH** variable, then this file is loaded into the current shell environment as if it were the argument to the **.** command except that only preset aliases are expanded, and a function of the given name is executed as described above. If not found, and the file **.paths** is found, and this file contains a line of the form **FPATH=path** where *path* names an existing directory, and this directory contains a file of the given name, then this file is loaded into the current shell environment as if it were the argument to the **.** special built-in command and a function of the given name is executed. Otherwise, if found, a process is created and an attempt is made to execute the command via *exec(2)*.

When an executable is found, the directory where it is found in is searched for a file named **.paths**. If this file is found and it contains a line of the form **BUILTIN_LIB=value**, then the library named by *value* will be searched for as if it were an option argument to **builtin -f**, and if it contains a built-in of the specified name this will be executed instead of a command by this name. Otherwise, if this file is found and it contains a line of the form *name=value* in the first or second line, then the environment variable *name* is modified by prepending the directory specified by *value* to the directory list. If *value* is not an absolute directory, then it specifies a directory relative to the directory that the executable was found. If the environment variable *name* does not already exist it will be added to the environment list for the specified command.

If the file has execute permission but is not an **a.out** file, it is assumed to be a file containing shell commands. A separate shell is spawned to read it. All non-exported variables are removed in this case. If the shell command file doesn't have read permission, or if the *setuid* and/or *setgid* bits are set on the file, then the shell executes an agent whose job it is to set up the permissions and execute the shell with the shell command file passed down as an open file. A parenthesized command is executed in a sub-shell without removing non-exported variables.

Command Re-entry.

The text of the last **HISTSIZE** (default 512) commands entered from a terminal device is saved in a *history* file. The file **\$HOME/sh_history** is used if the **HISTFILE** variable is not set or if the file it names is not writable. A shell can access the commands of all *interactive* shells which use the same named **HISTFILE**. The built-in command **hist** is used to list or edit a portion of this file. The portion of the file to be edited or listed can be selected by number or by giving the first character or characters of the command. A single command or range of commands can be specified. If you do not specify an editor program as an argument to **hist** then the value of the variable **HISTEDIT** is used. If **HISTEDIT** is unset, the obsolete variable **FCEDIT** is used. If **FCEDIT** is not defined, then **/bin/ed** is used. The edited command(s) is printed and re-executed upon leaving the editor unless you quit without writing. The **-s** option (and in obsolete versions, the editor name **-**) is used to skip the editing phase and to re-execute the command. In this case a substitution parameter of the form *old=new* can be used to modify the command before execution. For example, with the preset alias **r**, which is aliased to **'hist -s'**, typing **'r bad=good c'** will re-execute the most recent command which starts with the letter **c**, replacing the first occurrence of the string **bad** with the string **good**.

In-line Editing Options.

Normally, each command line entered from a terminal device is simply typed followed by a **new-line** ('RETURN' or 'LINE FEED'). If either the **emacs**, **gmacs**, or **vi** option is active, the user can edit the command line. To be in either of these edit modes **set** the corresponding option. An editing option is automatically selected each time the **VISUAL** or **EDITOR** variable is assigned a value ending in either of these option names.

The editing features require that the user's terminal accept 'RETURN' as carriage return without line feed and that a space (' ') must overwrite the current character on the screen.

Unless the **multiline** option is on, the editing modes implement a concept where the user is looking through a window at the current line. The window width is the value of **COLUMNS** if it is defined, otherwise 80. If the window width is too small to display the prompt and leave at least 8 columns to enter input, the prompt is truncated from the left. If the line is longer than the window width minus two, a mark is displayed at the end of the window to notify the user. As the cursor moves and reaches the window boundaries the window will be centered about the cursor. The mark is a > (<, *) if the line extends on the right (left, both) side(s) of the window.

The search commands in each edit mode provide access to the history file. Only strings are matched, not patterns, although a leading ^ in the string restricts the match to begin at the first character in the line.

Each of the edit modes has an operation to list the files or commands that match a partially entered word. When applied to the first word on the line, or the first word after a ;, |, &, or (, and the word does not begin with ~ or contain a /, the list of aliases, functions, and executable commands defined by the **PATH** variable that could match the partial word is displayed. Otherwise, the list of files that match the given word is displayed. If the partially entered word does not contain any file expansion characters, a * is appended before generating these lists. After displaying the generated list, the input line is redrawn. These operations are called command name listing and file name listing, respectively. There are additional operations, referred to as command name completion and file name completion, which compute the list of matching commands or files, but instead of printing the list, replace the current word with a complete or partial match. For file name completion, if the match is unique, a / is appended if the file is a directory and a space is appended if the file is not a directory. Otherwise, the longest common prefix for all the matching files replaces the word. For command name completion, only the portion of the file names after the last / are used to find the longest command prefix. If only a single name matches this prefix, then the word is replaced with the command name followed by a space. When using a *tab* for completion that does not yield a unique match, a subsequent *tab* will provide a numbered list of matching alternatives. A specific selection can be made by entering the selection number followed by a *tab*.

Key Bindings.

The **KEYBD** trap can be used to intercept keys as they are typed and change the characters that are actually seen by the shell. This trap is executed after each character (or sequence of characters when the first character is ESC) is entered while reading from a terminal. The variable **.sh.edchar** contains the character or character sequence which generated the trap. Changing the value of **.sh.edchar** in the trap action causes the shell to behave as if the new value were entered from the keyboard rather than the original value.

The variable **.sh.edcol** is set to the input column number of the cursor at the time of the input. The variable **.sh.edmode** is set to ESC when in **vi** insert mode (see below) and is null otherwise. By prepending **\${.sh.editmode}** to a value assigned to **.sh.edchar** it will cause the shell to change to control mode if it is not already in this mode.

This trap is not invoked for characters entered as arguments to editing directives, or while reading input for a character search.

Emacs Editing Mode.

This mode is entered by enabling either the **emacs** or **gmacs** option. The only difference between these two modes is the way they handle ^T. To edit, the user moves the cursor to the point needing correction and then inserts or deletes characters or words as needed. All the editing commands are control characters or escape sequences. The notation for control characters is caret (^) followed by the character. For example, ^F is the notation for control F. This is entered by depressing 'f' while holding down the 'CTRL'

(control) key. The ‘SHIFT’ key is *not* depressed. (The notation `^?` indicates the DEL (delete) key.)

The notation for escape sequences is **M-** followed by a character. For example, **M-f** (pronounced Meta f) is entered by depressing ESC (ascii **033**) followed by ‘f’. (**M-F** would be the notation for ESC followed by ‘SHIFT’ (capital) ‘F’.)

All edit commands operate from any place on the line (not just at the beginning). Neither the ‘RETURN’ nor the ‘LINE FEED’ key is entered after edit commands except when noted.

| | |
|-----------------|--|
| ^F | Move cursor forward (right) one character. |
| M-[C | Move cursor forward (right) one character. |
| M-f | Move cursor forward one word. (The emacs editor’s idea of a word is a string of characters consisting of only letters, digits and underscores.) |
| ^B | Move cursor backward (left) one character. |
| M-[D | Move cursor backward (left) one character. |
| M-b | Move cursor backward one word. |
| ^A | Move cursor to start of line. |
| M-[H | Move cursor to start of line. |
| ^E | Move cursor to end of line. |
| M-[Y | Move cursor to end of line. |
| ^]char | Move cursor forward to character <i>char</i> on current line. |
| M-^]char | Move cursor backward to character <i>char</i> on current line. |
| ^XX | Interchange the cursor and mark. |
| <i>erase</i> | (User defined erase character as defined by the <i>stty</i> (1) command, usually ^H or # .) Delete previous character. |
| <i>lnext</i> | (User defined literal next character as defined by the <i>stty</i> (1) command, or ^V if not defined.) Removes the next character’s editing features (if any). |
| ^D | Delete current character. |
| M-d | Delete current word. |
| M-^H | (Meta-backspace) Delete previous word. |
| M-h | Delete previous word. |
| M-^? | (Meta-DEL) Delete previous word (if your interrupt character is ^? (DEL, the default) then this command will not work). |
| ^T | Transpose current character with previous character and advance the cursor in <i>emacs</i> mode. Transpose two previous characters in <i>gmacs</i> mode. |
| ^C | Capitalize current character. |
| M-c | Capitalize current word. |
| M-l | Change the current word to lower case. |
| ^K | Delete from the cursor to the end of the line. If preceded by a numerical parameter whose value is less than the current cursor position, then delete from given position up to the cursor. If preceded by a numerical parameter whose value is greater than the current cursor position, then delete from cursor up to given cursor position. |
| ^W | Kill from the cursor to the mark. |
| M-p | Push the region from the cursor to the mark on the stack. |
| <i>kill</i> | (User defined kill character as defined by the <i>stty</i> command, usually ^G or @ .) Kill the entire current line. If two <i>kill</i> characters are entered in succession, all kill characters from then on cause a line feed (useful when using paper terminals). |
| ^Y | Restore last item removed from line. (Yank item back to the line.) |
| ^L | Line feed and print current line. |
| M-^L | Clear the screen. |
| ^@ | (Null character) Set mark. |
| M-space | (Meta space) Set mark. |
| ^J | (New line) Execute the current line. |
| ^M | (Return) Execute the current line. |
| <i>eof</i> | End-of-file character, normally ^D , is processed as an End-of-file only if the current line is null. |

| | |
|------------------|---|
| ^P | Fetch previous command. Each time ^P is entered the previous command back in time is accessed. Moves back one line when not on the first line of a multi-line command. |
| M-[A | Equivalent to ^P . |
| M-< | Fetch the least recent (oldest) history line. |
| M-> | Fetch the most recent (youngest) history line. |
| ^N | Fetch next command line. Each time ^N is entered the next command line forward in time is accessed. |
| M-[B | Equivalent to ^N . |
| ^Rstring | Reverse search history for a previous command line containing <i>string</i> . If a parameter of zero is given, the search is forward. <i>String</i> is terminated by a 'RETURN' or 'NEW LINE'. If <i>string</i> is preceded by a ^, the matched line must begin with <i>string</i> . If <i>string</i> is omitted, then the next command line containing the most recent <i>string</i> is accessed. In this case a parameter of zero reverses the direction of the search. |
| ^O | Operate – Execute the current line and fetch the next line relative to current line from the history file. |
| M-digits | (Escape) Define numeric parameter, the digits are taken as a parameter to the next command. The commands that accept a parameter are ^F , ^B , <i>erase</i> , ^C , ^D , ^K , ^R , ^P , ^N , ^] , M- , M-^] , M-^_ , M-= , M-b , M-c , M-d , M-f , M-h , M-l and M-^H . |
| M-letter | Soft-key – Your alias list is searched for an alias by the name <i>_letter</i> and if an alias of this name is defined, its value will be inserted on the input queue. The <i>letter</i> must not be one of the above meta-functions. |
| M-[letter | Soft-key – Your alias list is searched for an alias by the name <i>__letter</i> and if an alias of this name is defined, its value will be inserted on the input queue. This can be used to program function keys on many terminals. |
| M- | The last word of the previous command is inserted on the line. If preceded by a numeric parameter, the value of this parameter determines which word to insert rather than the last word. |
| M-^_ | Same as M- . |
| M-* | Attempt file name generation on the current word. An asterisk is appended if the word doesn't match any file or contain any special pattern characters. |
| M-ESC | Command or file name completion as described above. |
| ^I tab | Attempts command or file name completion as described above. If a partial completion occurs, repeating this will behave as if M-= were entered. If no match is found or entered after <i>space</i> , a <i>tab</i> is inserted. |
| M-= | If not preceded by a numeric parameter, it generates the list of matching commands or file names as described above. Otherwise, the word under the cursor is replaced by the item corresponding to the value of the numeric parameter from the most recently generated command or file list. If the cursor is not on a word, it is inserted instead. |
| ^U | Multiply parameter of next command by 4. |
| \ | Escape next character. Editing characters, the user's erase, kill and interrupt (normally ^?) characters may be entered in a command line or in a search string if preceded by a \ . The \ removes the next character's editing features (if any). |
| M-^V | Display version of the shell. |
| M-# | If the line does not begin with a # , a # is inserted at the beginning of the line and after each new-line, and the line is entered. This causes a comment to be inserted in the history file. If the line begins with a # , the # is deleted and one # after each new-line is also deleted. |

Vi Editing Mode.

There are two typing modes. Initially, when you enter a command you are in the *input* mode. To edit, the user enters *control* mode by typing ESC (**033**) and moves the cursor to the point needing correction and then inserts or deletes characters or words as needed. Most control commands accept an optional repeat *count* prior to the command.

When in **vi** mode on most systems, canonical processing is initially enabled and the command will be echoed again if the speed is 1200 baud or greater and it contains any control characters or less than one second has elapsed since the prompt was printed. The ESC character terminates canonical processing for the

remainder of the command and the user can then modify the command line. This scheme has the advantages of canonical processing with the type-ahead echoing of raw mode.

If the option **viraw** is also set, the terminal will always have canonical processing disabled. This mode is implicit for systems that do not support two alternate end of line delimiters, and may be helpful for certain terminals.

Input Edit Commands

By default the editor is in input mode.

| | |
|---------------|---|
| <i>erase</i> | (User defined erase character as defined by the <code>stty</code> command, usually ^H or #) Delete previous character. |
| ^W | Delete the previous blank separated word. On some systems the viraw option may be required for this to work. |
| <i>eof</i> | As the first character of the line causes the shell to terminate unless the ignoreeof option is set. Otherwise this character is ignored. |
| <i>lnext</i> | (User defined literal next character as defined by the <code>stty(1)</code> or ^V if not defined.) Removes the next character's editing features (if any). On some systems the viraw option may be required for this to work. |
| \ | Escape the next <i>erase</i> or <i>kill</i> character. |
| ^I tab | Attempts command or file name completion as described above and returns to input mode. If a partial completion occurs, repeating this will behave as if = were entered from control mode. If no match is found or entered after <i>space</i> , a <i>tab</i> is inserted. |

Motion Edit Commands

These commands will move the cursor.

| | |
|--------------------------|--|
| <i>[count]</i> l | Cursor forward (right) one character. |
| <i>[count]</i> [C | Cursor forward (right) one character. |
| <i>[count]</i> w | Cursor forward one alpha-numeric word. |
| <i>[count]</i> W | Cursor to the beginning of the next word that follows a blank. |
| <i>[count]</i> e | Cursor to end of word. |
| <i>[count]</i> E | Cursor to end of the current blank delimited word. |
| <i>[count]</i> h | Cursor backward (left) one character. |
| <i>[count]</i> [D | Cursor backward (left) one character. |
| <i>[count]</i> b | Cursor backward one word. |
| <i>[count]</i> B | Cursor to preceding blank separated word. |
| <i>[count]</i> | Cursor to column <i>count</i> . |
| <i>[count]</i> fc | Find the next character <i>c</i> in the current line. |
| <i>[count]</i> Fc | Find the previous character <i>c</i> in the current line. |
| <i>[count]</i> tc | Equivalent to f followed by h . |
| <i>[count]</i> Tc | Equivalent to F followed by l . |
| <i>[count]</i> ; | Repeats <i>count</i> times, the last single character find command, f , F , t , or T . |
| <i>[count]</i> , | Reverses the last single character find command <i>count</i> times. |
| 0 | Cursor to start of line. |
| ^ | Cursor to start of line. |
| [H | Cursor to first non-blank character in line. |
| \$ | Cursor to end of line. |
| [Y | Cursor to end of line. |
| % | Moves to balancing (,), { , }, [, or]. If cursor is not on one of the above characters, the remainder of the line is searched for the first occurrence of one of the above characters first. |

Search Edit Commands

These commands access your command history.

| | |
|--------------------------|--|
| <i>[count]</i> k | Fetch previous command. Each time k is entered the previous command back in time is accessed. |
| <i>[count]</i> - | Equivalent to k . |
| <i>[count]</i> [A | Equivalent to k . |

| | |
|-------------------|--|
| [count]j | Fetch next command. Each time j is entered the next command forward in time is accessed. |
| [count]+ | Equivalent to j . |
| [count][B] | Equivalent to j . |
| [count]G | The command number <i>count</i> is fetched. The default is the least recent history command. |
| /string | Search backward through history for a previous command containing <i>string</i> . <i>String</i> is terminated by a 'RETURN' or 'NEW LINE'. If <i>string</i> is preceded by a ^, the matched line must begin with <i>string</i> . If <i>string</i> is null, the previous string will be used. |
| ?string | Same as / except that search will be in the forward direction. |
| n | Search for next match of the last pattern to / or ? commands. |
| N | Search for next match of the last pattern to / or ? , but in reverse direction. |

Text Modification Edit Commands

These commands will modify the line.

a Enter input mode and enter text after the current character.

A Append text to the end of the line. Equivalent to **\$a**.

[count]c*motion*

c[*count*]*motion*

Delete current character through the character that *motion* would move the cursor to and enter input mode. If *motion* is **c**, the entire line will be deleted and input mode entered.

C Delete the current character through the end of line and enter input mode. Equivalent to **c\$**.

S Equivalent to **cc**.

[count]s Replace characters under the cursor in input mode.

D Delete the current character through the end of line. Equivalent to **d\$**.

[count]d*motion*

d[*count*]*motion*

Delete current character through the character that *motion* would move to. If *motion* is **d**, the entire line will be deleted.

i Enter input mode and insert text before the current character.

I Insert text before the beginning of the line. Equivalent to **0i**.

[count]P Place the previous text modification before the cursor.

[count]p Place the previous text modification after the cursor.

R Enter input mode and replace characters on the screen with characters you type overlay fashion.

[count]rc Replace the *count* character(s) starting at the current cursor position with *c*, and advance the cursor.

[count]x Delete current character.

[count]X Delete preceding character.

[count]. Repeat the previous text modification command.

[count]~ Invert the case of the *count* character(s) starting at the current cursor position and advance the cursor.

[count]_ Causes the *count* word of the previous command to be appended and input mode entered. The last word is used if *count* is omitted.

***** Causes an ***** to be appended to the current word and file name generation attempted. If no match is found, it rings the bell. Otherwise, the word is replaced by the matching pattern and input mode is entered.

**** Command or file name completion as described above.

Other Edit Commands

Miscellaneous commands.

[count]y*motion*

| | |
|---------------------------------------|--|
| y <i>[count]</i> <i>motion</i> | Yank current character through character that <i>motion</i> would move the cursor to and puts them into the delete buffer. The text and cursor are unchanged. |
| yy | Yanks the entire line. |
| Y | Yanks from current position to end of line. Equivalent to y\$. |
| u | Undo the last text modifying command. |
| U | Undo all the text modifying commands performed on the line. |
| [count]v | Returns the command hist -e \${VISUAL:-\${EDITOR:-vi}} <i>count</i> in the input buffer. If <i>count</i> is omitted, then the current line is used. |
| ^L | Line feed and print current line. Has effect only in control mode. |
| ^J | (New line) Execute the current line, regardless of mode. |
| ^M | (Return) Execute the current line, regardless of mode. |
| # | If the first character of the command is a #, then this command deletes this # and each # that follows a newline. Otherwise, sends the line after inserting a # in front of each line in the command. Useful for causing the current line to be inserted in the history as a comment and uncommenting previously commented commands in the history file. |
| [count]= | If <i>count</i> is not specified, it generates the list of matching commands or file names as described above. Otherwise, the word under the the cursor is replaced by the <i>count</i> item from the most recently generated command or file list. If the cursor is not on a word, it is inserted instead. |
| @letter | Your alias list is searched for an alias by the name <i>_letter</i> and if an alias of this name is defined, its value will be inserted on the input queue for processing. |
| ^V | Display version of the shell. |

Built-in Commands.

The following simple-commands are executed in the shell process. Input/Output redirection is permitted. Unless otherwise indicated, the output is written on file descriptor 1 and the exit status, when there is no syntax error, is zero. Except for **:**, **true**, **false**, **echo**, **newgrp**, and **login**, all built-in commands accept **--** to indicate end of options. They also interpret the option **--man** as a request to display the man page onto standard error and **-?** as a help request which prints a *usage* message on standard error. Commands that are preceded by one or two † symbols are special built-in commands and are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words following a command preceded by †† that are in the format of a variable assignment are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and field splitting and file name generation are not performed.

† : [*arg* ...]

The command only expands parameters.

† .*name* [*arg* ...]

If *name* is a function defined with the **function** *name* reserved word syntax, the function is executed in the current environment (as if it had been defined with the *name()* syntax.) Otherwise if *name* refers to a file, the file is read in its entirety and the commands are executed in the current shell environment. The search path specified by **PATH** is used to find the directory containing the file. If any arguments *arg* are given, they become the positional parameters while processing the . command and the original positional parameters are restored upon completion. Otherwise the positional parameters are unchanged. The exit status is the exit status of the last command executed.

†† **alias** [**-ptx**] [*name* [*=value*]] ...

alias with no arguments prints the list of aliases in the form *name=value* on standard output. The **-p** option causes the word **alias** to be inserted before each one. When one or more arguments are

given, an *alias* is defined for each *name* whose *value* is given. A trailing space in *value* causes the next word to be checked for alias substitution. The obsolete **-t** option is used to set and list tracked aliases. The value of a tracked alias is the full pathname corresponding to the given *name*. The value becomes undefined when the value of **PATH** is reset but the alias remains tracked. Without the **-t** option, for each *name* in the argument list for which no *value* is given, the name and value of the alias is printed. The obsolete **-x** option has no effect. The exit status is non-zero if a *name* is given, but no value, and no alias has been defined for the *name*.

bg [*job*...]

This command is only on systems that support job control. Puts each specified *job* into the background. The current job is put in the background if *job* is not specified. See *Jobs* for a description of the format of *job*.

† break [*n*]

Exit from the enclosing **for**, **while**, **until**, or **select** loop, if any. If *n* is specified, then break *n* levels.

builtin [**-ds**] [**-f file**] [*name*...]

If *name* is not specified, and no **-f** option is specified, the built-ins are printed on standard output. The **-s** option prints only the special built-ins. Otherwise, each *name* represents the pathname whose basename is the name of the built-in. The entry point function name is determined by prepending **b_** to the built-in name. The ISO C/C++ prototype is **b_mycommand(int argc, char *argv[], void *context)** for the builtin command *mycommand* where *argv* is array an of *argc* elements and context is an optional pointer to a **Shell_t** structure as described in **<ast/shell.h>**.

Special built-ins cannot be bound to a pathname or deleted. The **-d** option deletes each of the given built-ins. On systems that support dynamic loading, the **-f** option names a shared library containing the code for built-ins. The shared library prefix and/or suffix, which depend on the system, can be omitted. Once a library is loaded, its symbols become available for subsequent invocations of **builtin**. Multiple libraries can be specified with separate invocations of the **builtin** command. Libraries are searched in the reverse order in which they are specified. When a library is loaded, it looks for a function in the library whose name is **lib_init()** and invokes this function with an argument of **0**.

cd [**-LP**] [*arg*]

cd [**-LP**] *old new*

This command can be in either of two forms. In the first form it changes the current directory to *arg*. If *arg* is **-** the directory is changed to the previous directory. The shell variable **HOME** is the default *arg*. The variable **PWD** is set to the current directory. The shell variable **CDPATH** defines the search path for the directory containing *arg*. Alternative directory names are separated by a colon (:). The default path is **<null>** (specifying the current directory). Note that the current directory is specified by a null path name, which can appear immediately after the equal sign or between the colon delimiters anywhere else in the path list. If *arg* begins with a / then the search path is not used. Otherwise, each directory in the path is searched for *arg*.

The second form of **cd** substitutes the string *new* for the string *old* in the current directory name, **PWD**, and tries to change to this new directory.

By default, symbolic link names are treated literally when finding the directory name. This is equivalent to the **-L** option. The **-P** option causes symbolic links to be resolved when determining the directory. The last instance of **-L** or **-P** on the command line determines which method is used.

The **cd** command may not be executed by **rksh**. **rksh93**.

command [**-pvxV**] *name* [*arg*...]

Without the **-v** or **-V** options, **command** executes *name* with the arguments given by *arg*. The **-p** option causes a default path to be searched rather than the one defined by the value of **PATH**. Functions will not be searched for when finding *name*. In addition, if *name* refers to a special

built-in, none of the special properties associated with the leading daggers will be honored. (For example, the predefined alias **redirect=command exec** prevents a script from terminating when an invalid redirection is given.) With the **-x** option, if command execution would result in a failure because there are too many arguments, **errno E2BIG**, the shell will invoke command *name* multiple times with a subset of the arguments on each invocation. Arguments that occur prior to the first word that expands to multiple arguments and after the last word that expands to multiple arguments will be passed on each invocation. The exit status will be the maximum invocation exit status. With the **-v** option, **command** is equivalent to the built-in **whence** command described below. The **-V** option causes **command** to act like **whence -v**.

† **continue** [*n*]

Resume the next iteration of the enclosing **for**, **while**, **until**, or **select** loop. If *n* is specified, then resume at the *n*-th enclosing loop.

disown [*job*...]

Causes the shell not to send a HUP signal to each given *job*, or all active jobs if *job* is omitted, when a login shell terminates.

echo [*arg*...]

When the first *arg* does not begin with a **-**, and none of the arguments contain a ****, then **echo** prints each of its arguments separated by a space and terminated by a new-line. Otherwise, the behavior of **echo** is system dependent and **print** or **printf** described below should be used. See *echo(1)* for usage and description.

† **eval** [*arg*...]

The arguments are read as input to the shell and the resulting command(s) executed.

† **exec** [**-c**] [**-a** *name*] [*arg*...]

If *arg* is given, the command specified by the arguments is executed in place of this shell without creating a new process. The **-c** option causes the environment to be cleared before applying variable assignments associated with the **exec** invocation. The **-a** option causes *name* rather than the first *arg*, to become **argv[0]** for the new process. Input/output arguments may appear and affect the current process. If *arg* is not given, the effect of this command is to modify file descriptors as prescribed by the input/output redirection list. In this case, any file descriptor numbers greater than 2 that are opened with this mechanism are closed when invoking another program.

† **exit** [*n*]

Causes the shell to exit with the exit status specified by *n*. The value will be the least significant 8 bits of the specified status. If *n* is omitted, then the exit status is that of the last command executed. An end-of-file will also cause the shell to exit except for a shell which has the **ignoreeof** option (see **set** below) turned on.

†† **export** [**-p**] [*name*[=*value*]]...

If *name* is not given, the names and values of each variable with the export attribute are printed with the values quoted in a manner that allows them to be re-input. The **-p** option causes the word **export** to be inserted before each one. Otherwise, the given *names* are marked for automatic export to the *environment* of subsequently-executed commands.

false Does nothing, and exits 1. Used with **until** for infinite loops.

fg [*job*...]

This command is only on systems that support job control. Each *job* specified is brought to the foreground and waited for in the specified order. Otherwise, the current job is brought into the foreground. See *Jobs* for a description of the format of *job*.

getconf [*name* [*pathname*]]

Prints the current value of the configuration parameter given by *name*. The configuration parameters are defined by the IEEE POSIX 1003.1 and IEEE POSIX 1003.2 standards. (See *pathconf(2)* and *sysconf(2)*.) The *pathname* argument is required for parameters whose value depends on the location in the file system. If no arguments are given, **getconf** prints the names and values of the

current configuration parameters. The pathname */* is used for each of the parameters that requires *pathname*.

getopts [**-a** *name*] *optstring* *vname* [*arg* ...]

Checks *arg* for legal options. If *arg* is omitted, the positional parameters are used. An option argument begins with a + or a -. An option not beginning with + or - or the argument -- ends the options. Options beginning with + are only recognized when *optstring* begins with a +. *optstring* contains the letters that **getopts** recognizes. If a letter is followed by a :, that option is expected to have an argument. The options can be separated from the argument by blanks. The option -? causes **getopts** to generate a usage message on standard error. The -a argument can be used to specify the name to use for the usage message, which defaults to \$0.

getopts places the next option letter it finds inside variable *vname* each time it is invoked. The option letter will be prepended with a + when *arg* begins with a +. The index of the next *arg* is stored in **OPTIND**. The option argument, if any, gets stored in **OPTARG**.

A leading : in *optstring* causes **getopts** to store the letter of an invalid option in **OPTARG**, and to set *vname* to ? for an unknown option and to : when a required option argument is missing. Otherwise, **getopts** prints an error message. The exit status is non-zero when there are no more options.

There is no way to specify any of the options :, +, -, ?, [, and]. The option # can only be specified as the first option.

hist [**-e** *ename*] [**-nlr**] [*first* [*last*]]

hist -s [*old=new*] [*command*]

In the first form, a range of commands from *first* to *last* is selected from the last **HISTSIZE** commands that were typed at the terminal. The arguments *first* and *last* may be specified as a number or as a string. A string is used to locate the most recent command starting with the given string. A negative number is used as an offset to the current command number. If the -l option is selected, the commands are listed on standard output. Otherwise, the editor program *ename* is invoked on a file containing these keyboard commands. If *ename* is not supplied, then the value of the variable **HISTEDIT** is used. If **HISTEDIT** is not set, then **FCEDIT** (default **/bin/ed**) is used as the editor. When editing is complete, the edited command(s) is executed if the changes have been saved. If *last* is not specified, then it will be set to *first*. If *first* is not specified, the default is the previous command for editing and -16 for listing. The option -r reverses the order of the commands and the option -n suppresses command numbers when listing. In the second form, *command* is interpreted as *first* described above and defaults to the last command executed. The resulting command is executed after the optional substitution *old=new* is performed.

jobs [**-lnp**] [*job* ...]

Lists information about each given job; or all active jobs if *job* is omitted. The -l option lists process ids in addition to the normal information. The -n option only displays jobs that have stopped or exited since last notified. The -p option causes only the process group to be listed. See *Jobs* for a description of the format of *job*.

kill [**-s** *signame*] *job* ...

kill [**-n** *signum*] *job* ...

kill -l [*sig* ...]

Sends either the TERM (terminate) signal or the specified signal to the specified jobs or processes. Signals are either given by number with the -n option or by name with the -s option (as given in <signal.h>, stripped of the prefix "SIG" with the exception that SIGCLD is named CHLD). For backward compatibility, the n and s can be omitted and the number or name placed immediately after the -. If the signal being sent is TERM (terminate) or HUP (hangup), then the job or process will be sent a CONT (continue) signal if it is stopped. The argument *job* can be the process id of a process that is not a member of one of the active jobs. See *Jobs* for a description of the format of *job*. In the third form, **kill -l**, if *sig* is not specified, the signal names are listed. Otherwise, for each *sig* that is a name, the corresponding signal number is listed. For each *sig* that is a number,

the signal name corresponding to the least significant 8 bits of *sig* is listed.

let *arg* ...

Each *arg* is a separate *arithmetic expression* to be evaluated. See *Arithmetic Evaluation* above, for a description of arithmetic expression evaluation.

The exit status is 0 if the value of the last expression is non-zero, and 1 otherwise.

† **newgrp** [*arg* ...]

Equivalent to **exec /bin/newgrp** *arg* ...

print [**-Renprs**] [**-u** *unit*] [**-f** *format*] [*arg* ...]

With no options or with option **-** or **--**, each *arg* is printed on standard output. The **-f** option causes the arguments to be printed as described by **printf**. In this case, any **e**, **n**, **r**, **R** options are ignored. Otherwise, unless the **-R** or **-r**, are specified, the following escape conventions will be applied:

| | |
|------------|---|
| \a | The alert character (ascii 07). |
| \b | The backspace character (ascii 010). |
| \c | Causes print to end without processing more arguments and not adding a new-line. |
| \f | The formfeed character (ascii 014). |
| \n | The new-line character (ascii 012). |
| \r | The carriage return character (ascii 015). |
| \t | The tab character (ascii 011). |
| \v | The vertical tab character (ascii 013). |
| \E | The escape character (ascii 033). |
| \ | The backslash character \ . |
| \0x | The character defined by the 1, 2, or 3-digit octal string given by <i>x</i> . |

The **-R** option will print all subsequent arguments and options other than **-n**. The **-e** causes the above escape conventions to be applied. This is the default behavior. It reverses the effect of an earlier **-r**. The **-p** option causes the arguments to be written onto the pipe of the process spawned with **|&** instead of standard output. The **-s** option causes the arguments to be written onto the history file instead of standard output. The **-u** option can be used to specify a one digit file descriptor unit number *unit* on which the output will be placed. The default is 1. If the option **-n** is used, no **new-line** is added to the output.

printf *format* [*arg* ...]

The arguments *arg* are printed on standard output in accordance with the ANSI-C formatting rules associated with the format string *format*. If the number of arguments exceeds the number of format specifications, the **format** string is reused to format remaining arguments. The following extensions can also be used: A **%b** format can be used instead of **%s** to cause escape sequences in the corresponding *arg* to be expanded as described in **print**. A **%B** option causes each of the arguments to be treated as variable names and the binary value of variable will be printed. This is most useful for variables whose attribute is **-b**. A **%H** format can be used instead of **%s** to cause characters in *arg* that are special in HTML and XML to be output as their entity name. A **%P** format can be used instead of **%s** to cause *arg* to be interpreted as an extended regular expression and be printed as a shell pattern. A **%R** format can be used instead of **%s** to cause *arg* to be interpreted as a shell pattern and to be printed as an extended regular expression. A **%q** format can be used instead of **%s** to cause the resulting string to be quoted in a manner than can be reinput to the shell. A **%(date-format)T** format can be used to treat an argument as a date/time string and to format the date/time according to the *date-format* as defined for the **date(1)** command. A **%Z** format will output a byte whose value is 0. The precision field of the **%d** format can be followed by a **.** and the output base. In this case, the **#** flag character causes *base#* to be prepended. The **#** flag when used with the **d** specifier without an output base, causes the output to be displayed in thousands units with one of the suffixes **k M G T P E** to indicate the unit. The **#** flag when used with the **i** specifier causes the output to be displayed in 1024 with one of the suffixes **Ki Mi Gi Ti Pi Ei** to indicate the unit. The **=** flag has been added to center the output within the specified field width.

pwd [**-LP**]

Outputs the value of the current working directory. The **-L** option is the default; it prints the logical name of the current directory. If the **-P** option is given, all symbolic links are resolved from the name. The last instance of **-L** or **-P** on the command line determines which method is used.

read [**-Aprs**] [**-d delim**] [**-n n**] [[**-N n**] [[**-t timeout**] [**-u unit**] [*vname*?*prompt*] [*vname* ...]

The shell input mechanism. One line is read and is broken up into fields using the characters in **IFS** as separators. The escape character, ****, is used to remove any special meaning for the next character and for line continuation. The **-d** option causes the read to continue to the first character of *delim* rather than new-line. The **-n** option causes at most *n* bytes to read rather a full line but will return when reading from a slow device as soon as any characters have been read. The **-N** option causes exactly *n* to be read unless an end-of-file has been encountered or the read times out because of the **-t** option. In raw mode, **-r**, the **** character is not treated specially. The first field is assigned to the first *vname*, the second field to the second *vname*, etc., with leftover fields assigned to the last *vname*. When *vname* has the binary attribute and **-n** or **-N** is specified, the bytes that are read are stored directly into the variable. If the **-v** is specified, then the value of the first *vname* will be used as a default value when reading from a terminal device. The **-A** option causes the variable *vname* to be unset and each field that is read to be stored in successive elements of the indexed array *vname*. The **-p** option causes the input line to be taken from the input pipe of a process spawned by the shell using **|&**. If the **-s** option is present, the input will be saved as a command in the history file. The option **-u** can be used to specify a one digit file descriptor unit *unit* to read from. The file descriptor can be opened with the **exec** special built-in command. The default value of unit *n* is 0. The option **-t** is used to specify a timeout in seconds when reading from a terminal or pipe. If *vname* is omitted, then **REPLY** is used as the default *vname*. An end-of-file with the **-p** option causes cleanup for this process so that another can be spawned. If the first argument contains a **?**, the remainder of this word is used as a *prompt* on standard error when the shell is interactive. The exit status is 0 unless an end-of-file is encountered or **read** has timed out.

†† **readonly** [**-p**] [*vname*[=*value*]] ...

If *vname* is not given, the names and values of each variable with the readonly attribute is printed with the values quoted in a manner that allows them to be re-inputted. The **-p** option causes the word **readonly** to be inserted before each one. Otherwise, the given *vnames* are marked readonly and these names cannot be changed by subsequent assignment.

† **return** [*n*]

Causes a shell *function* or **.** script to return to the invoking script with the exit status specified by *n*. The value will be the least significant 8 bits of the specified status. If *n* is omitted, then the return status is that of the last command executed. If **return** is invoked while not in a *function* or a **.** script, then it behaves the same as **exit**.

† **set** [**±CGabefhkmnoprstuvx**] [**±o** [*option*]] ... [**±A** *vname*] [*arg* ...]

The options for this command have meaning as follows:

- A** Array assignment. Unset the variable *vname* and assign values sequentially from the *arg* list. If **+A** is used, the variable *vname* is not unset first.
- B** Enable brace pattern field generation. This is the default behavior.
- C** Prevents redirection **>** from truncating existing files. Files that are created are opened with the **O_EXCL** mode. Requires **>|** to truncate a file when turned on.
- G** Causes the pattern ****** by itself to match files and zero or more directories and sub-directories when used for file name generation. If followed by a **/** only directories and sub-directories are matched.
- a** All subsequent variables that are defined are automatically exported.
- b** Prints job completion messages as soon as a background job changes state rather than waiting for the next prompt.
- e** If a command has a non-zero exit status, execute the **ERR** trap, if set, and exit. This mode is disabled while reading profiles.

- f** Disables file name generation.
- h** Each command becomes a tracked alias when first encountered.
- k** (Obsolete). All variable assignment arguments are placed in the environment for a command, not just those that precede the command name.
- m** Background jobs will run in a separate process group and a line will print upon completion. The exit status of background jobs is reported in a completion message. On systems with job control, this option is turned on automatically for interactive shells.
- n** Read commands and check them for syntax errors, but do not execute them. Ignored for interactive shells.
- o** The following argument can be one of the following option names:
 - allexport** Same as **-a**.
 - errexit** Same as **-e**.
 - bgnice** All background jobs are run at a lower priority. This is the default mode.
 - braceexpand** Same as **-B**.
 - emacs** Puts you in an *emacs* style in-line editor for command entry.
 - globstar** Same as **-G**.
 - gmacs** Puts you in a *gmacs* style in-line editor for command entry.
 - ignoreeof** The shell will not exit on end-of-file. The command **exit** must be used.
 - keyword** Same as **-k**.
 - markdirs** All directory names resulting from file name generation have a trailing **/** appended.
 - monitor** Same as **-m**.
 - multiline** The built-in editors will use multiple lines on the screen for lines that are longer than the width of the screen. This may not work for all terminals.
 - noclobber** Same as **-C**.
 - noexec** Same as **-n**.
 - noglob** Same as **-f**.
 - nolog** Do not save function definitions in the history file.
 - notify** Same as **-b**.
 - nounset** Same as **-u**.
 - pipefail** A pipeline will not complete until all components of the pipeline have completed, and the return value will be the value of the last non-zero command to fail or zero if no command has failed.
 - showme** When enabled, simple commands or pipelines preceded by a semicolon (**;**) will be displayed as if the **xtrace** option were enabled but will not be executed. Otherwise, the leading **;** will be ignored.
 - privileged** Same as **-p**.
 - verbose** Same as **-v**.
 - trackall** Same as **-h**.
 - vi** Puts you in insert mode of a *vi* style in-line editor until you hit the escape character **033**. This puts you in control mode. A return sends the line.
 - viraw** Each character is processed as it is typed in *vi* mode.
 - xtrace** Same as **-x**.
- p** Disables processing of the **\$HOME/.profile** file and uses the file **/etc/suid_profile** instead of the **ENV** file. This mode is on whenever the effective uid (gid) is not equal to

the real uid (gid). Turning this off causes the effective uid and gid to be set to the real uid and gid.

- r** Enables the restricted shell. This option cannot be unset once set.
- s** Sort the positional parameters lexicographically.
- t** (Obsolete). Exit after reading and executing one command.
- u** Treat unset parameters as an error when substituting.
- v** Print shell input lines as they are read.
- x** Print commands and their arguments as they are executed.
- Do not change any of the options; useful in setting **\$1** to a value beginning with **-**. If no arguments follow this option then the positional parameters are unset.

As an obsolete feature, if the first *arg* is **-** then the **-x** and **-v** options are turned off and the next *arg* is treated as the first argument. Using **+** rather than **-** causes these options to be turned off. These options can also be used upon invocation of the shell. The current set of options may be found in **\$-**. Unless **-A** is specified, the remaining arguments are positional parameters and are assigned, in order, to **\$1 \$2 ...**. If no arguments are given, then the names and values of all variables are printed on the standard output.

† **shift** [*n*]

The positional parameters from **\$n+1 ...** are renamed **\$1 ...**, default *n* is 1. The parameter *n* can be any arithmetic expression that evaluates to a non-negative number less than or equal to **\$#**.

sleep *seconds*

Suspends execution for the number of decimal seconds or fractions of a second given by *seconds*.

† **trap** [**-p**] [*action*] [*sig*] ...

The **-p** option causes the trap action associated with each trap as specified by the arguments to be printed with appropriate quoting. Otherwise, *action* will be processed as if it were an argument to **eval** when the shell receives signal(s) *sig*. Each *sig* can be given as a number or as the name of the signal. Trap commands are executed in order of signal number. Any attempt to set a trap on a signal that was ignored on entry to the current shell is ineffective. If *action* is omitted and the first *sig* is a number, or if *action* is **-**, then the trap(s) for each *sig* are reset to their original values. If *action* is the null string then this signal is ignored by the shell and by the commands it invokes. If *sig* is **ERR** then *action* will be executed whenever a command has a non-zero exit status. If *sig* is **DEBUG** then *action* will be executed before each command. The variable **.sh.command** will contain the contents of the current command line when *action* is running. If *sig* is **0** or **EXIT** and the **trap** statement is executed inside the body of a function defined with the **function name** syntax, then the command *action* is executed after the function completes. If *sig* is **0** or **EXIT** for a **trap** set outside any function then the command *action* is executed on exit from the shell. If *sig* is **KEYBD**, then *action* will be executed whenever a key is read while in **emacs**, **gmacs**, or **vi** mode. The **trap** command with no arguments prints a list of commands associated with each signal number.

true Does nothing, and exits 0. Used with **while** for infinite loops.

†† **typeset** [**±AHflabnprtux**] [**±EFLRZi**[*n*]] [*vname*[*=value*]] ...

Sets attributes and values for shell variables and functions. When invoked inside a function defined with the **function name** syntax, a new instance of the variable *vname* is created, and the variable's value and type are restored when the function completes. The following list of attributes may be specified:

- A** Declares *vname* to be an associative array. Subscripts are strings rather than arithmetic expressions.
- a** Declares *vname* to be an indexed array. This is optional unless except for compound variable assignments.
- E** Declares *vname* to be a double precision floating point number. If *n* is non-zero, it defines the number of significant figures that are used when expanding *vname*. Otherwise, ten significant figures will be used.

- F** Declares *vname* to be a double precision floating point number. If *n* is non-zero, it defines the number of places after the decimal point that are used when expanding *vname*. Otherwise ten places after the decimal point will be used.
- H** This option provides UNIX to host-name file mapping on non-UNIX machines.
- L** Left justify and remove leading blanks from *value*. If *n* is non-zero, it defines the width of the field, otherwise it is determined by the width of the value of first assignment. When the variable is assigned to, it is filled on the right with blanks or truncated, if necessary, to fit into the field. The **-R** option is turned off.
- R** Right justify and fill with leading blanks. If *n* is non-zero, it defines the width of the field, otherwise it is determined by the width of the value of first assignment. The field is left filled with blanks or truncated from the end if the variable is reassigned. The **-L** option is turned off.
- Z** Right justify and fill with leading zeros if the first non-blank character is a digit and the **-L** option has not been set. Remove leading zeros if the **-L** option is also set. If *n* is non-zero, it defines the width of the field, otherwise it is determined by the width of the value of first assignment.
- f** The names refer to function names rather than variable names. No assignments can be made and the only other valid options are **-t**, **-u** and **-x**. The **-t** option turns on execution tracing for this function. The **-u** option causes this function to be marked undefined. The **FPATH** variable will be searched to find the function definition when the function is referenced. If no options other than **-f** is specified, then the function definition will be displayed on standard output. If **+f** is specified, then a line containing the function name followed by a shell comment containing the line number and path name of the file where this function was defined, if any, is displayed.
- b** The variable can hold any number of bytes of data. The data can be text or binary. The value is represented by the base64 encoding of the data. If **-Z** is also specified, the size in bytes of the data in the buffer will be determined by the size associated with the **-Z**. If the base64 string assigned results in more data, it will be truncated. Otherwise, it will be filled with bytes whose value is zero. The **printf** format **%B** can be used to output the actual data in this buffer instead of the base64 encoding of the data.
- i** Declares *vname* to be represented internally as integer. The right hand side of an assignment is evaluated as an arithmetic expression when assigning to an integer. If *n* is non-zero, it defines the output arithmetic base, otherwise the output base will be ten.
- l** All upper-case characters are converted to lower-case. The upper-case option, **-u**, is turned off.
- n** Declares *vname* to be a reference to the variable whose name is defined by the value of variable *vname*. This is usually used to reference a variable inside a function whose name has been passed as an argument.
- r** The given *vnames* are marked readonly and these names cannot be changed by subsequent assignment.
- t** Tags the variables. Tags are user definable and have no special meaning to the shell.
- u** All lower-case characters are converted to upper-case. The lower-case option, **-l**, is turned off.
- x** The given *vnames* are marked for automatic export to the *environment* of subsequently-executed commands. Variables whose names contain a **.** cannot be exported.

The **-i** attribute cannot be specified along with **-R**, **-L**, **-Z**, or **-f**.

Using **+** rather than **-** causes these options to be turned off. If no *vname* arguments are given, a list of *vnames* (and optionally the *values*) of the *variables* is printed. (Using **+** rather than **-** keeps the values from being printed.) The **-p** option causes **typeset** followed by the option letters to be printed before each name rather than the names of the options. If any option other than **-p** is given, only those variables which have all of the given options are printed. Otherwise, the *vnames* and *attributes* of all *variables* that have attributes are printed.

ulimit [**-HSacdfmnpstv**] [*limit*]

Set or display a resource limit. The available resource limits are listed below. Many systems do not support one or more of these limits. The limit for a specified resource is set when *limit* is specified. The value of *limit* can be a number in the unit specified below with each resource, or the value **unlimited**. The **-H** and **-S** options specify whether the hard limit or the soft limit for the given resource is set. A hard limit cannot be increased once it is set. A soft limit can be increased up to the value of the hard limit. If neither the **H** nor **S** option is specified, the limit applies to both. The current resource limit is printed when *limit* is omitted. In this case, the soft limit is printed unless **H** is specified. When more than one resource is specified, then the limit name and unit is printed before the value.

- a** Lists all of the current resource limits.
- c** The number of 512-byte blocks on the size of core dumps.
- d** The number of K-bytes on the size of the data area.
- f** The number of 512-byte blocks on files that can be written by the current process or by child processes (files of any size may be read).
- m** The number of K-bytes on the size of physical memory.
- n** The number of file descriptors plus 1.
- p** The number of 512-byte blocks for pipe buffering.
- s** The number of K-bytes on the size of the stack area.
- t** The number of CPU seconds to be used by each process.
- v** The number of K-bytes for virtual memory.

If no option is given, **-f** is assumed.

umask [**-S**] [*mask*]

The user file-creation mask is set to *mask* (see *umask(2)*). *mask* can either be an octal number or a symbolic value as described in *chmod(1)*. If a symbolic value is given, the new umask value is the complement of the result of applying *mask* to the complement of the previous umask value. If *mask* is omitted, the current value of the mask is printed. The **-S** option causes the mode to be printed as a symbolic value. Otherwise, the mask is printed in octal.

† **unalias** [**-a**] *name* ...

The aliases given by the list of *names* are removed from the alias list. The **-a** option causes all the aliases to be unset.

† **unset** [**-fnv**] *vname* ...

The variables given by the list of *vnames* are unassigned, i.e., their values and attributes are erased. Readonly variables cannot be unset. If the **-f** option is set, then the names refer to *function* names. If the **-v** option is set, then the names refer to *variable* names. The **-f** option overrides **-v**. If **-n** is set and *name* is a name reference, then *name* will be unset rather than the variable that it references. The default is equivalent to **-v**. Unsetting **LINENO**, **MAILCHECK**, **OPTARG**, **OPTIND**, **RANDOM**, **SECONDS**, **TMOUT**, and **_** removes their special meaning even if they are subsequently assigned to.

wait [*job* ...]

Wait for the specified *job* and report its termination status. If *job* is not given, then all currently active child processes are waited for. The exit status from this command is that of the last process waited for if *job* is specified; otherwise it is zero. See *Jobs* for a description of the format of *job*.

whence [**-afpv**] *name* ...

For each *name*, indicate how it would be interpreted if used as a command name.

The **-v** option produces a more verbose report. The **-f** option skips the search for functions. The **-p** option does a path search for *name* even if *name* is an alias, a function, or a reserved word. The **-p** option turns off the **-v** option. The **-a** option is similar to the **-v** option but causes all interpretations of the given name to be reported.

Invocation.

If the shell is invoked by *exec*(2), and the first character of argument zero (**\$0**) is **-**, then the shell is assumed to be a *login* shell and commands are read from **/etc/profile** and then from either **.profile** in the current directory or **\$HOME/.profile**, if either file exists. Next, for interactive shells, commands are read from the file named by performing parameter expansion, command substitution, and arithmetic substitution on the value of the environment variable **ENV** if the file exists. If the **-s** option is not present and *arg* and a file by the name of *arg* exists, then it reads and executes this script. Otherwise, if the first *arg* does not contain a */*, a path search is performed on the first *arg* to determine the name of the script to execute. The script *arg* must have execute permission and any *setuid* and *setgid* settings will be ignored. If the script is not found on the path, *arg* is processed as if it named a built-in command or function. Commands are then read as described below; the following options are interpreted by the shell when it is invoked:

- c** If the **-c** option is present, then commands are read from the first *arg*. Any remaining arguments become positional parameters starting at **0**.
- s** If the **-s** option is present or if no arguments remain, then commands are read from the standard input. Shell output, except for the output of the *Special Commands* listed above, is written to file descriptor 2.
- i** If the **-i** option is present or if the shell input and output are attached to a terminal (as told by *tcgetattr*(2)), then this shell is *interactive*. In this case **TERM** is ignored (so that **kill 0** does not kill an interactive shell) and **INTR** is caught and ignored (so that **wait** is interruptible). In all cases, **QUIT** is ignored by the shell.
- r** If the **-r** option is present, the shell is a restricted shell.
- D** A list of all double quoted strings that are preceded by a **\$** will be printed on standard output and the shell will exit. This set of strings will be subject to language translation when the locale is not C or POSIX. No commands will be executed.
- P** If **-P** or **-o profile** is present, the shell is a profile shell (see *pfexec*(1)).
- R filename** The **-R filename** option is used to generate a cross reference database that can be used by a separate utility to find definitions and references for variables and commands.

The remaining options and arguments are described under the **set** command above. An optional **-** as the first argument is ignored.

Rksh Only.

Rksh is used to set up login names and execution environments whose capabilities are more controlled than those of the standard shell. The actions of **rksh** are identical to those of **ksh**, except that the following are disallowed:

- Unsetting the restricted option.
- changing directory (see *cd*(1)),
- setting or unsetting the value or attributes of **SHELL**, **ENV**, **FPATH**, or **PATH**,
- specifying path or command names containing */*,
- redirecting output (**>**, **>|**, **<>**, and **>>**).
- adding or deleting built-in commands.
- using **command -p** to invoke a command.

The restrictions above are enforced after **.profile** and the **ENV** files are interpreted.

When a command to be executed is found to be a shell procedure, **rksh** invokes *ksh* to execute it. Thus, it is possible to provide to the end-user shell procedures that have access to the full power of the standard shell, while imposing a limited menu of commands; this scheme assumes that the end-user does not have write and execute permissions in the same directory.

The net effect of these rules is that the writer of the **.profile** has complete control over user actions, by performing guaranteed setup actions and leaving the user in an appropriate directory (probably *not* the login directory).

The system administrator often sets up a directory of commands (e.g., **/usr/rbin**) that can be safely invoked by **rksh**.

EXIT STATUS

Errors detected by the shell, such as syntax errors, cause the shell to return a non-zero exit status. If the shell is being used non-interactively, then execution of the shell file is abandoned unless the error occurs inside a subshell in which case the subshell is abandoned. Otherwise, the shell returns the exit status of the last command executed (see also the **exit** command above). Run time errors detected by the shell are reported by printing the command or function name and the error condition. If the line number that the error occurred on is greater than one, then the line number is also printed in square brackets (**[]**) after the command or function name.

FILES

/etc/profile The system wide initialization file, executed for login shells.

\$HOME/.profile

The personal initialization file, executed for login shells after **/etc/profile**.

\$HOME/.kshrc

Default personal initialization file, executed for interactive shells when **ENV** is not set.

/etc/suid_profile

Alternative initialization file, executed when instead of personal initialization file when the real and effective user or group id do not match.

/dev/null

NULL device

SEE ALSO

cat(1), **cd(1)**, **chmod(1)**, **cut(1)**, **egrep(1)**, **echo(1)**, **emacs(1)**, **env(1)**, **fgrep(1)**, **gmacs(1)**, **grep(1)**, **newgrp(1)**, **pfexec(1)**, **stty(1)**, **test(1)**, **umask(1)**, **vi(1)**, **dup(2)**, **exec(2)**, **fork(2)**, **getpwnam(3)**, **ioctl(2)**, **lseek(2)**, **paste(1)**, **pathconf(2)**, **pipe(2)**, **sysconf(2)**, **umask(2)**, **ulimit(2)**, **wait(2)**, **rand(3)**, **a.out(5)**, **profile(5)**, **environ(7)**.

Morris I. Bolsky and David G. Korn, *The New KornShell Command and Programming Language*, Prentice Hall, 1995.

POSIX – Part 2: Shell and Utilities, IEEE Std 1003.2-1992, ISO/IEC 9945-2, IEEE, 1993.

CAVEATS

If a command is executed, and then a command with the same name is installed in a directory in the search path before the directory where the original command was found, the shell will continue to *exec* the original command. Use the **-t** option of the **alias** command to correct this situation.

Some very old shell scripts contain a **^** as a synonym for the pipe character **|**.

Using the **hist** built-in command within a compound command will cause the whole command to disappear from the history file.

The built-in command **.file** reads the whole file before any commands are executed. Therefore, **alias** and **unalias** commands in the file will not apply to any commands defined in the file.

Traps are not processed while a job is waiting for a foreground process. Thus, a trap on **CHLD** won't be executed until the foreground job terminates.

It is a good idea to leave a space after the comma operator in arithmetic expressions to prevent the comma from being interpreted as the decimal point character in certain locales.

NAME

ksh_wrapper – wrap Korn- and Bourne Again Shell scripts in a Bourne shell

SYNOPSIS

edrc/bin/kshscript -> .ksh_wrapper

edrc/lib/kshscript -> .ksh_wrapper

edrc/sbin/kshscript -> .ksh_wrapper

edrc/lib/kshell

AVAILABILITY

WA2L/edrc

DESCRIPTION

wrap Korn- and Bourne Again shell scripts to avoid using `#!/bin/ksh` in the magic key. However, due to the fact that some scripts need Korn shell functionality either **ksh** or **bash** has to be installed in `/bin/`, `/sbin/`, `/usr/contrib/bin/` or `/usr/local/bin/`.

Further **ksh** / **bash** locations can be configured in the config file **ksh_wrapper.cfg**.

To start a new command thru the **.ksh_wrapper**, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/sbin/**, **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ][*edrc*/bash]: ln -s .ksh_wrapper new_cmd
```

- 2.) install the ksh script in the **edrc/lib/ksh/** directory:

```
[ /opt/edrc/lib/ksh ]
[ root@acme001 ][*edrc*/bash]: vi new_cmd

[ /opt/edrc/lib/ksh ]
[ root@acme001 ][*edrc*/bash]: chmod 644 new_cmd
```

The command **edrc/lib/kshell** returns the path of the Korn- or Bourne Again shell available on the system. The path resolution and configuration file is the same as in **.ksh_wrapper**.

OPTIONS

edrc/lib/kshell

print path to **ksh** or **bash**.

edrc/bin/.ksh_wrapper

wrapper for **ksh** / **bash** scripts in **edrc/bin/**.

edrc/lib/.ksh_wrapper

wrapper for **ksh** / **bash** scripts in **edrc/lib/**.

ENVIRONMENT

-

EXIT STATUS

- | | |
|------------|--|
| 101 | shell (ksh or bash) not found. If ksh or bash is installed on the system and you get this error, add the shell location in the configuration file ksh_wrapper.cfg . |
| 102 | the Korn-Shell script to be started that should be located in lib/ksh/ does not exist. |
| 103 | user calling the command has no permission to access/execute the called <i>daemon_command</i> . |
| 106 | configuration file ksh_wrapper.cfg does not exist. |
| 107 | the .ksh_wrapper was called directly. |

FILES

etc/ksh_wrapper.cfg

configuration file for the **.ksh_wrapper** command.

lib/ksh/

location of the wrapped Korn-/Bourne Again shell scripts. This files should have the file permissions *644* to show, that those scripts must not be started directly.

EXAMPLES

-

SEE ALSO

edrcintro(1), **bash(1)**, **cmdlist(1m)**, **daemon_wrapper(1)**, **ksh(1)**, **java_wrapper(1)**, **ksh_wrapper.cfg(4)**, **lua_wrapper(1)**, **ln(1)**, **osid(3)**, **os_wrapper(1)**, **perl_wrapper(1)**

NOTES

-

BUGS

-

AUTHOR

ksh_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

kshrc – shell initialization file

SYNOPSIS

edrc/etc/kshrc

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the main configuration file (rc-file) for the **shell**(1), the **eshell**(1) and the **edrc** '**shell**' command.

Do not edit this file, it will be patched with the WA2L/edrc file; define your customization and settings to the **etc/shell.cfg** file.

FILEFORMAT

The fileformat is a ksh/bash script file.

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **bash**(1), **edrc**(1m), **eshell**(1), **ksh**(1), **shell**(1), **shell.cfg**(4)

NOTES

-

BUGS

-

AUTHOR

kshrc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ksh_wrapper – wrap Korn- and Bourne Again Shell scripts in a Bourne shell

SYNOPSIS

edrc/bin/kshscript -> **.ksh_wrapper**

edrc/lib/kshscript -> **.ksh_wrapper**

edrc/sbin/kshscript -> **.ksh_wrapper**

edrc/lib/kshell

AVAILABILITY

WA2L/edrc

DESCRIPTION

wrap Korn- and Bourne Again shell scripts to avoid using `#!/bin/ksh` in the magic key. However, due to the fact that some scripts need Korn shell functionality either **ksh** or **bash** has to be installed in `/bin/`, `/sbin/`, `/usr/contrib/bin/` or `/usr/local/bin/`.

Further **ksh** / **bash** locations can be configured in the config file **ksh_wrapper.cfg**.

To start a new command thru the **.ksh_wrapper**, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/sbin/**, **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ][*edrc*/bash]: ln -s .ksh_wrapper new_cmd
```

- 2.) install the ksh script in the **edrc/lib/ksh/** directory:

```
[ /opt/edrc/lib/ksh ]
[ root@acme001 ][*edrc*/bash]: vi new_cmd

[ /opt/edrc/lib/ksh ]
[ root@acme001 ][*edrc*/bash]: chmod 644 new_cmd
```

The command **edrc/lib/kshell** returns the path of the Korn- or Bourne Again shell available on the system. The path resolution and configuration file is the same as in **.ksh_wrapper**.

OPTIONS

edrc/lib/kshell

print path to **ksh** or **bash**.

edrc/bin/.ksh_wrapper

wrapper for **ksh** / **bash** scripts in **edrc/bin/**.

edrc/lib/.ksh_wrapper

wrapper for **ksh** / **bash** scripts in **edrc/lib/**.

ENVIRONMENT

-

EXIT STATUS

- | | |
|------------|--|
| 101 | shell (ksh or bash) not found. If ksh or bash is installed on the system and you get this error, add the shell location in the configuration file ksh_wrapper.cfg . |
| 102 | the Korn-Shell script to be started that should be located in lib/ksh/ does not exist. |
| 103 | user calling the command has no permission to access/execute the called <i>daemon_command</i> . |
| 106 | configuration file ksh_wrapper.cfg does not exist. |
| 107 | the .ksh_wrapper was called directly. |

FILES

etc/ksh_wrapper.cfg

configuration file for the **.ksh_wrapper** command.

lib/ksh/

location of the wrapped Korn-/Bourne Again shell scripts. This files should have the file permissions *644* to show, that those scripts must not be started directly.

EXAMPLES

-

SEE ALSO

edrcintro(1), **bash(1)**, **cmdlist(1m)**, **daemon_wrapper(1)**, **ksh(1)**, **java_wrapper(1)**, **ksh_wrapper.cfg(4)**, **lua_wrapper(1)**, **ln(1)**, **osid(3)**, **os_wrapper(1)**, **perl_wrapper(1)**

NOTES

-

BUGS

-

AUTHOR

ksh_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ksh_wrapper.cfg – configuration file for .ksh_wrapper

SYNOPSIS

edrc/etc/ksh_wrapper.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **.ksh_wrapper** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SEARCH_PATH**

Colon separated search path where to search for the **ksh** or **bash** interpreter.

The path defined here is prepended to the default.

Example: SEARCH_PATH="/bin:/sbin:/usr/local/bin"

Default: SEARCH_PATH="/bin:/sbin:/usr/local/bin"

SEE ALSO

edrcintro(1), ksh_wrapper(1)

NOTES

-

BUGS

-

AUTHOR

ksh_wrapper.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`leo` – commandline interface to <http://dict.leo.org/>.

SYNOPSIS

```
leo [-slmcfuphdv] [<term>]
```

DESCRIPTION

leo is a commandline interface to the german/english/french dictionary on <http://dict.leo.org/>. It supports almost all features which the website supports, plus more.

Results will be printed to the terminal. By default the searched key word will be highlighted (which can be turned off, see below).

To get faster results, **leo** is able to cache queries if you repeatedly use the same query.

leo acts as a standard webbrowser as your mozilla or what so ever does, it connects to the website, executes the query, parses the HTML result and finally prints it somewhat nicely formatted to the terminal.

As of this writing **leo** acts as:

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.8.1.9) Gecko/20071025 Firefox/1.
```

OPTIONS

`-s --spelltolerance`

Allow spelling errors.

Possible values: **standard**, **on** or **off**.

Default setting: **standard**.

`-m --morphology`

Provide morphology information.

Possible values: **standard**, **none** or **forcedAll**.

Default setting: **standard**.

`-c --chartolerance`

Allow umlaut alternatives.

Possible values: **fuzzy**, **exact** or **relaxed**.

Default: **relaxed**.

`-l --language`

Translation direction.

Possible values: **en**, **fr**, **de2en**, **en2de**, **de2fr** or **fr2de**.

en and **fr** do select the translation direction automatically.

Default: **en**.

`-n --noescapechars`

Don't use escapes for highlighting.

Default: do highlighting.

Controllable via config file too. See below.

No highlighting will be used if STDOUT is not connected to a terminal.

`-f --force`

Don't use the query cache.

Default: use the cache.

This option has no effect if **use_cache** is turned off in the config file.

- `-u --user`
Specify the http proxy user to use if your proxy requires authentication. Read the 'PROXY' section for more details.
- `-p --passwd`
Specify the cleartext password to use with http proxy authentication.
This is not recommended and just implemented for completeness.
- `-h --help`
Display this help and exit.
- `-v --version`
Display version information and exit.
- `-d --debug`
Enable debugging output (a lot of it, beware!), which will be printed to STDERR. If you find a bug you must supply the debugging output along with your bugreport.

term is the key word which you want to translate. If the term contains white spaces quote it using double quotes.

If the **term** parameter is not specified, **leo** will read it from STDIN.

CONFIG

leo reads a config file **.leo** in your home directory if it exists. The following variables are supported:

use_latin

Turns on conversion of UTF8 characters to their latin* encoding.

Default setting (if not given): **yes**.

use_cache

Controls the use of the cache (see later).

Possible values: **yes** or **no**.

Default setting(if not given): **yes**.

If the commandline option **-f** or **--force** has been set then the cache will not be used for the query and if for this query exists an entry in the cache it will be removed from it.

use_color

Controls the use of escape sequences in the terminal output to highlight the key-word in the result.

Possible values: **yes** or **no**.

Default setting(if not given): **yes**.

You can set this option via commandline too: **-n** or **--noescapechars**.

The config option has higher precedence.

user_agent

You may modify the user agent as **leo** identifies itself on the target site. The default is:

```
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.3.1; X11)
```

CACHING

leo supports caching of queries for faster results if you repeatedly use the same query. A query consists of the given **term** (the key word or string) plus the translation option settings.

If you, for example, execute once the following query:

```
% leo langnase
```

and somewhere later:

```
% leo -c exact
```

then **leo** will treat the latter query as a different one than the previous one, because *dict.leo.org* behaves different when different translation options are given.

PROXY

leo can be used with a HTTP proxy service. For this to work, you only have to set the environment variable **http_proxy**. It has the following format:

```
PROTO://[USER:PASSWD@]SERVER[:PORT]
```

The only supported protocol is **http**. If your proxy works without authentication, you can omit the **user:passwd** part. If no port is specified, **80** will be used.

Here is an example (for bash):

```
export http_proxy=http://172.16.120.120:3128
```

and an example with authentication credentials:

```
export http_proxy=http://max:34dwe2@172.16.120.120:3128
```

As security is always important, I have to warn you, that other users on the same machine can read your environment using the 'ps -e ..' command, so this is not recommended.

The most secure way for proxy authentication is just to specify the server+port with **http_proxy** but no credentials, and instead use the **-u** commandline parameter to specify a user (do not use **-p** to specify the password, this will also be readyble in process listing). In this case, **leo** will ask you interactively for the password. It will try its best to hide it from being displayed when you type it (as most such routines in other tools do it as well), it this fails (e.g. because you do not have the 'stty' tool installed), the password will be read from STDIN.

FILES

```
~/.leo           the config file for leo. Not required.
~/.leo-CACHE.db* the cache file.
```

AUTHOR

Thomas Linden <tom@daemon.de>.

BUGS

leo depends on <http://dict.leo.org/>. It may break **leo** if they change something on the site. Therefore be so kind and inform me if you encounter some weird behavior of **leo**. In most cases it is not a bug of **leo** itself, it is a website change on <http://dict.leo.org/>.

In such a case repeat the failed query and use the commandline flag **-d** (which enables debugging) and send the full output to me, thanks.

COPYRIGHT

leo copyleft 2000–2014 Thomas Linden. All rights reserved.

<http://dict.leo.org/> copyright (c) 1995–2014 LEO Dictionary Team.

The search results returned by **leo** are based on the work of the people at LEO.org. Thanks for the great work.

Some time ago they told me that they are disagreed with **leo**, or in other words: from their point of view **leo** seems to break copyright law in some or another way.

I thought a long time about this, but I must deny this. **leo** acts as a simple web client, just like mozilla, IE or even lynx are doing. They are providing the service to the public so I use my favorite web browser to make use of it. In fact my favorite browser to view dict.leo.org is **leo**. There is nothing wrong with that. IMHO.

If you disagree or are asked by the LEO team to stop using **leo** you may decide this for yourself. I in my case wrote kinda browser, what is not prohibited. At least not today.

VERSION

This is the manpage for **leo** version **1.39**.

NAME

libexpect – programmed dialogue library with interactive programs

DESCRIPTION

This library contains functions that allow Expect to be used as a Tcl extension or to be used directly from C or C++ (without Tcl). Adding Expect as a Tcl extension is very short and simple, so that will be covered first.

SYNOPSIS

```
#include expect_tcl.h
Expect_Init(interp);
```

```
cc files... -lexpect5.20 -ltcl7.5 -lm
```

Note: library versions may differ in the actual release.

The Expect_Init function adds expect commands to the named interpreter. It avoids overwriting commands that already exist, however aliases beginning with "exp_" are always created for expect commands. So for example, "send" can be used as "exp_send".

Generally, you should only call Expect commands via Tcl_Eval. Certain auxiliary functions may be called directly. They are summarized below. They may be useful in constructing your own main. Look at the file exp_main_exp.c in the Expect distribution as a prototype main. Another prototype is tclAppInit.c in the Tcl source distribution. A prototype for working with Tk is in exp_main_tk.c in the Expect distribution.

```
int exp_cmdlinecmds;
int exp_interactive;
FILE *exp_cmdfile;
char *exp_cmdfilename;
int exp_tcl_debugger_available;

void exp_parse_argv(Tcl_Interp *,int argc,char **argv);
int exp_interpreter(Tcl_Interp *);
void exp_interpret_cmdfile(Tcl_Interp *,FILE *);
void exp_interpret_cmdfilename(Tcl_Interp *,char *);
void exp_interpret_rcfiles(Tcl_Interp *,int my_rc,int sys_rc);
char *exp_cook(char *s,int *len);
void (*exp_app_exit)EXP_PROTO((Tcl_Interp *));
void exp_exit(Tcl_Interp *,int status);
void exp_exit_handlers(Tcl_Interp *);
void exp_error(Tcl_Interp,char *,...);
```

exp_cmdlinecmds is 1 if Expect has been invoked with commands on the program command-line (using "-c" for example). **exp_interactive** is 1 if Expect has been invoked with the -i flag or if no commands or script is being invoked. **exp_cmdfile** is a stream from which Expect will read commands. **exp_cmdfilename** is the name of a file which Expect will open and read commands from. **exp_tcl_debugger_available** is 1 if the debugger has been armed.

exp_parse_argv reads the representation of the command line. Based on what is found, any of the other variables listed here are initialized appropriately. **exp_interpreter** interactively prompts the user for commands and evaluates them. **exp_interpret_cmdfile** reads the given stream and evaluates any commands found. **exp_interpret_cmdfilename** opens the named file and evaluates any commands found. **exp_interpret_rcfiles** reads and evaluates the .rc files. If my_rc is zero, then ~/.expectrc is skipped. If sys_rc is zero, then the system-wide expectrc file is skipped. **exp_cook** returns a static buffer containing the argument reproduced with newlines replaced by carriage-return linefeed sequences. The primary purpose of this is to allow messages to be produced without worrying about whether the terminal is in raw mode or cooked

mode. If length is zero, it is computed via strlen. **exp_error** is a printf-like function that writes the result to interp->result.

SYNOPSIS

```
#include <expect.h>
```

```
int
exp_spawnl(file, arg0 [, arg1, ..., argn] (char *)0);
char *file;
char *arg0, *arg1, ... *argn;
```

```
int
exp_spawnv(file,argv);
char *file, *argv[ ];
```

```
int
exp_spawnfd(fd);
int fd;
```

```
FILE *
exp_popen(command);
char *command;
```

```
extern int exp_pid;
extern int exp_ttyinit;
extern int exp_ttycopy;
extern int exp_console;
extern char *exp_stty_init;
extern void (*exp_close_in_child)();
extern void (*exp_child_exec_prelude)();
extern void exp_close_tcl_files();
```

```
cc files... -lexpect -ltcl -lm
```

DESCRIPTION

exp_spawnl and **exp_spawnv** fork a new process so that its stdin, stdout, and stderr can be written and read by the current process. *file* is the name of a file to be executed. The *arg* pointers are null-terminated strings. Following the style of `execve()`, *arg0* (or *argv[0]*) is customarily a duplicate of the name of the file.

Four interfaces are available, **exp_spawnl** is useful when the number of arguments is known at compile time. **exp_spawnv** is useful when the number of arguments is not known at compile time. **exp_spawnfd** is useful when an open file descriptor is already available as a source. **exp_popen** is explained later on.

If the process is successfully created, a file descriptor is returned which corresponds to the process's stdin, stdout and stderr. A stream may be associated with the file descriptor by using `fdopen()`. (This should almost certainly be followed by `setbuf()` to unbuffer the I/O.)

Closing the file descriptor will typically be detected by the process as an EOF. Once such a process exits, it should be waited upon (via `wait`) in order to free up the kernel process slot. (Some systems allow you to avoid this if you ignore the SIGCHLD signal).

exp_popen is yet another interface, styled after `popen()`. It takes a Bourne shell command line, and returns a stream that corresponds to the process's stdin, stdout and stderr. The actual implementation of **exp_popen** below demonstrates **exp_spawnl**.

```
FILE *
exp_popen(program)
```

```

char *program;
{
    FILE *fp;
    int ec;

    if (0 > (ec = exp_spawnl("sh", "sh", "-c", program, (char *)0)))
        return(0);
    if (NULL == (fp = fdopen(ec, "r+"))) return(0);
    setbuf(fp, (char *)0);
    return(fp);
}

```

After a process is started, the variable **exp_pid** is set to the process-id of the new process. The variable **exp_pty_slave_name** is set to the name of the slave side of the pty.

The spawn functions use a pty to communicate with the process. By default, the pty is initialized the same way as the user's tty (if possible, i.e., if the environment has a controlling terminal.) This initialization can be skipped by setting **exp_ttycopy** to 0.

The pty is further initialized to some system wide defaults if **exp_ttyinit** is non-zero. The default is generally comparable to "stty sane".

The tty setting can be further modified by setting the variable **exp_stty_init**. This variable is interpreted in the style of stty arguments. For example, **exp_stty_init** = "sane"; repeats the default initialization.

On some systems, it is possible to redirect console output to ptys. If this is supported, you can force the next spawn to obtain the console output by setting the variable **exp_console** to 1.

Between the time a process is started and the new program is given control, the spawn functions can clean up the environment by closing file descriptors. By default, the only file descriptors closed are ones internal to Expect and any marked "close-on-exec".

If needed, you can close additional file descriptors by creating an appropriate function and assigning it to **exp_close_in_child**. The function will be called after the fork and before the exec. (This also modifies the behavior of the spawn command in Expect.)

If you are also using Tcl, it may be convenient to use the function **exp_close_tcl_files** which closes all files between the default standard file descriptors and the highest descriptor known to Tcl. (Expect does this.)

The function **exp_child_exec_prelude** is the last function called prior to the actual exec in the child. You can redefine this for effects such as manipulating the uid or the signals.

IF YOU WANT TO ALLOCATE YOUR OWN PTY

```

extern int exp_autoalloctype;
extern int exp_pty[2];

```

The spawn functions use a pty to communicate with the process. By default, a pty is automatically allocated each time a process is spawned. If you want to allocate ptys yourself, before calling one of the spawn functions, set **exp_autoalloctype** to 0, **exp_pty[0]** to the master pty file descriptor and **exp_pty[1]** to the slave pty file descriptor. The expect library will not do any pty initializations (e.g., **exp_stty_init** will not be used). The slave pty file descriptor will be automatically closed when the process is spawned. After the process is started, all further communication takes place with the master pty file descriptor.

exp_spawnl and **exp_spawnv** duplicate the shell's actions in searching for an executable file in a list of directories. The directory list is obtained from the environment.

EXPECT PROCESSING

While it is possible to use `read()` to read information from a process spawned by `exp_spawnl` or `exp_spawnv`, more convenient functions are provided. They are as follows:

```
int
exp_expectl(fd,type1,pattern1,[re1,],value1,type2,...,exp_end);
int fd;
enum exp_type type;
char *pattern1, *pattern2, ...;
regex *re1, *re2, ...;
int value1, value2, ...;
```

```
int
exp_fexpectl(fp,type1,pattern1,[re1,]value1,type2,...,exp_end);
FILE *fp;
enum exp_type type;
char *pattern1, *pattern2, ...;
regex *re1, *re2, ...;
int value1, value2, ...;
```

```
enum exp_type {
    exp_end,
    exp_glob,
    exp_exact,
    exp_regex,
    exp_compiled,
    exp_null,
};
```

```
struct exp_case {
    char *pattern;
    regex *re;
    enum exp_type type;
    int value;
};
```

```
int
exp_expectv(fd,cases);
int fd;
struct exp_case *cases;
```

```
int
exp_fexpectv(fp,cases);
FILE *fp;
struct exp_case *cases;
```

```
extern int exp_timeout;
extern char *exp_match;
extern char *exp_match_end;
extern char *exp_buffer;
extern char *exp_buffer_end;
extern int exp_match_max;
extern int exp_full_buffer;
extern int exp_remove_nulls;
```


The functions wait until the output from a process matches one of the patterns, a specified time period has passed, or an EOF is seen.

The first argument to each function is either a file descriptor or a stream. Successive sets of arguments describe patterns and associated integer values to return when the pattern matches.

The type argument is one of four values. `exp_end` indicates that no more patterns appear. `exp_glob` indicates that the pattern is a glob-style string pattern. `exp_exact` indicates that the pattern is an exact string. `exp_regexp` indicates that the pattern is a regexp-style string pattern. `exp_compiled` indicates that the pattern is a regexp-style string pattern, and that its compiled form is also provided. `exp_null` indicates that the pattern is a null (for debugging purposes, a string pattern must also follow).

If the compiled form is not provided with the functions `exp_expectl` and `exp_fexpectl`, any pattern compilation done internally is thrown away after the function returns. The functions `exp_expectv` and `exp_fexpectv` will automatically compile patterns and will not throw them away. Instead, they must be discarded by the user, by calling `free` on each pattern. It is only necessary to discard them, the last time the cases are used.

Regexp subpatterns matched are stored in the compiled regexp. Assuming "re" contains a compiled regexp, the matched string can be found in `re->startp[0]`. The match substrings (according to the parentheses) in the original pattern can be found in `re->startp[1]`, `re->startp[2]`, and so on, up to `re->startp[9]`. The corresponding strings ends are `re->endp[x]` where x is that same index as for the string start.

The type `exp_null` matches if a null appears in the input. The variable `exp_remove_nulls` must be set to 0 to prevent nulls from being automatically stripped. By default, `exp_remove_nulls` is set to 1 and nulls are automatically stripped.

`exp_expectv` and `exp_fexpectv` are useful when the number of patterns is not known in advance. In this case, the sets are provided in an array. The end of the array is denoted by a struct `exp_case` with type `exp_end`. For the rest of this discussion, these functions will be referred to generically as *expect*.

If a pattern matches, then the corresponding integer value is returned. Values need not be unique, however they should be positive to avoid being mistaken for `EXP_EOF`, `EXP_TIMEOUT`, or `EXP_FULLBUFFER`. Upon EOF or timeout, the value `EXP_EOF` or `EXP_TIMEOUT` is returned. The default timeout period is 10 seconds but may be changed by setting the variable `exp_timeout`. A value of -1 disables a timeout from occurring. A value of 0 causes the expect function to return immediately (i.e., poll) after one `read()`. However it must be preceded by a function such as `select`, `poll`, or an event manager callback to guarantee that there is data to be read.

If the variable `exp_full_buffer` is 1, then `EXP_FULLBUFFER` is returned if `exp_buffer` fills with no pattern having matched.

When the expect function returns, `exp_buffer` points to the buffer of characters that was being considered for matching. `exp_buffer_end` points to one past the last character in `exp_buffer`. If a match occurred, `exp_match` points into `exp_buffer` where the match began. `exp_match_end` points to one character past where the match ended.

Each time new input arrives, it is compared to each pattern in the order they are listed. Thus, you may test for absence of a match by making the last pattern something guaranteed to appear, such as a prompt. In situations where there is no prompt, you must check for `EXP_TIMEOUT` (just like you would if you were interacting manually). More philosophy and strategies on specifying **expect** patterns can be found in the documentation on the **expect** program itself. See SEE ALSO below.

Patterns are the usual C-shell-style regular expressions. For example, the following fragment looks for a successful login, such as from a telnet dialogue.

```
switch (exp_expectl(
    exp_glob,"connected",CONN,
    exp_glob,"busy",BUSY,
```

```

        exp_glob,"failed",ABORT,
        exp_glob,"invalid password",ABORT,
        exp_end)) {
    case CONN:      /* logged in successfully */
        break;
    case BUSY:      /* couldn't log in at the moment */
        break;
    case EXP_TIMEOUT:
    case ABORT:     /* can't log in at any moment! */
        break;
    default: /* problem with expect */
    }

```

Asterisks (as in the example above) are a useful shorthand for omitting line-termination characters and other detail. Patterns must match the entire output of the current process (since the previous read on the descriptor or stream). More than 2000 bytes of output can force earlier bytes to be "forgotten". This may be changed by setting the variable **exp_match_max**. Note that excessively large values can slow down the pattern matcher.

RUNNING IN THE BACKGROUND

```

extern int exp_disconnected;
int exp_disconnect();

```

It is possible to move a process into the background after it has begun running. A typical use for this is to read passwords and then go into the background to sleep before using the passwords to do real work.

To move a process into the background, fork, call `exp_disconnect()` in the child process and `exit()` in the parent process. This disassociates your process from the controlling terminal. If you wish to move a process into the background in a different way, you must set the variable `exp_disconnected` to 1. This allows processes spawned after this point to be started correctly.

MULTIPLEXING

By default, the expect functions block inside of a read on a single file descriptor. If you want to wait on patterns from multiple file descriptors, use `select`, `poll`, or an event manager. They will tell you what file descriptor is ready to read.

When a file descriptor is ready to read, you can use the expect functions to do one and only read by setting timeout to 0.

SLAVE CONTROL

```

void
exp_slave_control(fd,enable)
int fd;
int enable;

```

Pty trapping is normally done automatically by the expect functions. However, if you want to issue an `ioctl`, for example, directly on the slave device, you should temporarily disable trapping.

Pty trapping can be controlled with `exp_slave_control`. The first argument is the file descriptor corresponding to the spawned process. The second argument is a 0 if trapping is to be disabled and 1 if it is to be enabled.

ERRORS

All functions indicate errors by returning -1 and setting `errno`.

Errors that occur after the spawn functions fork (e.g., attempting to spawn a non-existent program) are

written to the process's stderr, and will be read by the first **expect**.

SIGNALS

```
extern int exp_reading;
extern jmp_buf exp_readenv;
```

expect uses alarm() to timeout, thus if you generate alarms during **expect**, it will timeout prematurely.

Internally, **expect** calls read() which can be interrupted by signals. If you define signal handlers, you can choose to restart or abort **expect**'s internal read. The variable, **exp_reading**, is true if (and only if) **expect**'s read has been interrupted. longjmp(exp_readenv,EXP_ABORT) will abort the read. longjmp(exp_readenv,EXP_RESTART) will restart the read.

LOGGING

```
extern int exp_loguser;
extern int exp_logfile_all
extern FILE *exp_logfile;
```

If **exp_loguser** is nonzero, **expect** sends any output from the spawned process to stdout. Since interactive programs typically echo their input, this usually suffices to show both sides of the conversation. If **exp_logfile** is also nonzero, this same output is written to the stream defined by **exp_logfile**. If **exp_logfile_all** is non-zero, **exp_logfile** is written regardless of the value of **exp_loguser**.

DEBUGGING

While I consider the library to be easy to use, I think that the standalone expect program is much, much, easier to use than working with the C compiler and its usual edit, compile, debug cycle. Unlike typical C programs, most of the debugging isn't getting the C compiler to accept your programs - rather, it is getting the dialogue correct. Also, translating scripts from expect to C is usually not necessary. For example, the speed of interactive dialogues is virtually never an issue. So please try the standalone 'expect' program first. I suspect it is a more appropriate solution for most people than the library.

Nonetheless, if you feel compelled to debug in C, here are some tools to help you.

```
extern int exp_is_debugging;
extern FILE *exp_debugfile;
```

While expect dialogues seem very intuitive, trying to codify them in a program can reveal many surprises in a program's interface. Therefore a variety of debugging aids are available. They are controlled by the above variables, all 0 by default.

Debugging information internal to **expect** is sent to stderr when **exp_is_debugging** is non-zero. The debugging information includes every character received, and every attempt made to match the current input against the patterns. In addition, non-printable characters are translated to a printable form. For example, a control-C appears as a caret followed by a C. If **exp_logfile** is non-zero, this information is also written to that stream.

If **exp_debugfile** is non-zero, all normal and debugging information is written to that stream, regardless of the value of **exp_is_debugging**.

CAVEATS

The stream versions of the **expect** functions are much slower than the file descriptor versions because there is no way to portably read an unknown number of bytes without the potential of timing out. Thus, characters are read one at a time. You are therefore strongly encouraged to use the file descriptor versions of **expect** (although, automated versions of interactive programs don't usually demand high speed anyway).

You can actually get the best of both worlds, writing with the usual stream functions and reading with the file descriptor versions of **expect** as long as you don't attempt to intermix other stream input functions (e.g., fgetc). To do this, pass fileno(stream) as the file descriptor each time. Fortunately, there is little reason to

use anything but the **expect** functions when reading from interactive programs.

There is no matching `exp_pclose` to `exp_popen` (unlike `popen` and `pclose`). It only takes two functions to close down a connection (`fclose()` followed by waiting on the `pid`), but it is not uncommon to separate these two actions by large time intervals, so the function seems of little value.

If you are running on a Cray running Unicos (all I know for sure from experience), you must run your compiled program as `root` or `setuid`. The problem is that the Cray only allows `root` processes to open `ptys`. You should observe as much precautions as possible: If you don't need permissions, `setuid(0)` only immediately before calling one of the `spawn` functions and immediately set it back afterwards.

Normally, **spawn** takes little time to execute. If you notice `spawn` taking a significant amount of time, it is probably encountering `ptys` that are wedged. A number of tests are run on `ptys` to avoid entanglements with errant processes. (These take 10 seconds per wedged `pty`.) Running `expect` with the `-d` option will show if **expect** is encountering many `ptys` in odd states. If you cannot kill the processes to which these `ptys` are attached, your only recourse may be to reboot.

BUGS

The **exp_fexpect** functions don't work at all under HP-UX - it appears to be a bug in `getc`. Follow the advice (above) about using the **exp_expect** functions (which doesn't need to call `getc`). If you fix the problem (before I do - please check the latest release) let me know.

SEE ALSO

An alternative to this library is the **expect** program. **expect** interprets scripts written in a high-level language which direct the dialogue. In addition, the user can take control and interact directly when desired. If it is not absolutely necessary to write your own C program, it is much easier to use **expect** to perform the entire interaction. It is described further in the following references:

"expect: Curing Those Uncontrollable Fits of Interactivity" by Don Libes, Proceedings of the Summer 1990 USENIX Conference, Anaheim, California, June 11-15, 1990.

"Using expect to Automate System Administration Tasks" by Don Libes, Proceedings of the 1990 USENIX Large Installation Systems Administration Conference, Colorado Springs, Colorado, October 17-19, 1990.

`expect(1)`, `alarm(3)`, `read(2)`, `write(2)`, `fdopen(3)`, `execve(2)`, `execvp(3)`, `longjmp(3)`, `pty(4)`.

There are several examples C programs in the test directory of **expect**'s source distribution which use the `expect` library.

AUTHOR

Don Libes, `libes@nist.gov`, National Institute of Standards and Technology

ACKNOWLEDGEMENTS

Thanks to John Ousterhout (UCBerkeley) for supplying the pattern matcher.

Design and implementation of the **expect** library was paid for by the U.S. government and is therefore in the public domain. However the author and NIST would like credit if this program and documentation or portions of them are used.

NAME

lbanner – make posters in large letters

SYNOPSIS

edrc/lib/lbanner [**-h**]

lbanner [**-s** *size*] [**-r** *rotate*] [**-c** *character*] **-t** "*banner text*"

AVAILABILITY

WA2L/edrc

DESCRIPTION

print an ASCII banner as the **banner**(3) command does, but allow more options, as character size, fill character and banner orientation.

OPTIONS

-h usage message.

-s *size* size of the letters (**1** ... **5**).

-r *rotate* orientation of the banner (**v** = vertical, **h** = horizontal).

-c *character* character to be used to build the banner letters.

-t "*banner text*"
string to be printed in large letters. The qotes are only needed when a multiple word string has to be printed.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO**edrcintro(1)**, **banner(3)****NOTES**

-

BUGS

-

AUTHOR

lbanner was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lgcheckd – daemon to check, classify and report logfile content

SYNOPSIS

edrc/bin/lgcheckd [**-h** | **-V**]

lgcheckd -a (**start** | **stop** | **status**)

lgcheckd -a (**defined** | **monitored**)

lgcheckd -a ([no]collect | [no]ifstart)

lgcheckd -a (**refresh** | **reset**)

lgcheckd -a report

cat logfile | **lgcpattern** [**-s**] *patternfile*

cat logfile | **lgcpattern -p**

AVAILABILITY

WA2L/edrc

DESCRIPTION

NOTE: The **lgcheckd** daemon is a more efficient and leaner version of the **logcheckd(1m)** command. It is recommended to use **lgcheckd** for new setups.

With **lgcheckd** logfiles can be analyzed and the content can be classified to different levels of importance.

During analysis the timestamp when a certain log entry has been discovered is added to the collected analysis output.

The levels used to classify the logfile content can be defined in pattern files which are related to one or more logfiles to be analyzed.

Files listed in the **LOGFILE OPTIONS** section of the configuration file must not exist at start of the **lgcheckd**. However, as soon as such a file appears it is monitored.

The generated HTML 'Log File Check' report allows easy navigation in the report and provides a view of all matched log file entries per level of importance for a reporting period.

Only existing files or files that once existed during a reporting period will be listed in the report.

The HTML report is also suited for viewing with the **lynx(1)** browser.

lgcheckd is able to communicate to other commands for each matched logfile entry via the **INTERFACES** mechanism, which can be configured in the **lgcheckd.cfg** configuration file. Using interfaces, it is therefore

possible to forward discovered matches to a monitoring system that might not have a very sophisticated logfile analysis functionality.

lgcheckd will not fill up the file system where the log data is collected. If file space gets short during log file analysis the log file match data collection is suspended until sufficient disk space is available again (and will then resume automatically).

The suspension applies as soon as the free space is lower then **20** times the current size of the **state.db**, whereas this number can be changed in the config file using the **SUSPENDTRIGGER=number** setting. During the suspension the log files are still analysed, the matches are counted and the interfaces are started.

OPTIONS

-h usage message.

-V print daemon version.

-a action:

start start the log check daemon **lgcheckd** in the background. It is possible to start one daemon per system.

stop stop a running **lgcheckd** daemon.

report create a 'Log File Check' report for the period since the last report generation.

When invoking the **report** action, the current log collection run is ended, the log match data is deleted (depending on the **CLEAN_COLLECT** setting in the configuration file) and a new collection run is started.

All log files ever monitored are listed in the report.

However, if a log file has no matches in the current log collection run, '**<no pattern matches>**' is shown in the **Logfiles** section of the report.

When logfiles have matches, but during it's occurrence the log file match collection was suspended, the related match counts are shown in the **Overview** section of the report. In that case in the **Logfiles** section '**<no data available>**' is shown as long during the whole collect run duration no other matches were recorded.

The produced report file name is printed plain to **stdout** to allow easy capturing and further processing (as mailing the report using **mail_file(1)**).

reset initialize the **lgcheckd** daemon. All state information, collected logfile content match data, all counters and trend information will be cleared.

status check if a **lgcheckd** is running (EXIT STATUS: **14**), not running (EXIT STATUS: **15**) or is shutting down (EXIT STATUS **18**) on the system.

defined

list the logfiles defined in the **lgcheckd.cfg** (respectively **logcheckd.cfg**) config file.

monitored

list the currently monitored logfiles including additional information of the current- and past collection runs.

When the **lgcheckd** experiences heavy load due to many log entries that have to be analysed, it switches temporarily from **SINGLE** to **BURST** write mode. Whereas in **SINGLE** mode match counts and log matches are immediately written to the database, in **BURST** mode the daemon delays writing to the database and then writes the information in larger chunks to it. This to increase performance.

Therefore the information shown when using **-a monitored** is not up to date while the daemon is in **BURST** mode.

nocollect

disable collecting of matched logfile data. The matches are nevertheless counted when the collection is disabled.

collect enable collecting of matched logfile data.

noifstart

disable the start of interfaces on related log file data match.

ifstart enable the start of interfaces on related log file data match.

refresh force the resolution of log files to monitor.

The interval of checking for new log files that appeared based on the definition in the configuration file is **5** minutes.

lgcpattern *patternfile*

utility to help to define, verify and optimize a **logcheckd.pattern(4)** file. This utility replaces the similar functionality '**logcheckd -a verify patternfile check_levels**' as provided by **logcheckd(1m)**.

See **lgcpattern(3)** for more information.

ENVIRONMENT

-

EXIT STATUS

0 no error.

- 1** neither configfile **etc/lgcheckd.cfg** nor **etc/logcheckd.cfg** exists.
- 4** usage listed.
- 5** program version printed.
- 14** daemon running.
- 15** daemon not running.
- 18** daemon is shutting down.

FILES

etc/lgcheckd.cfg

configuration file of **lgcheckd**, see **lgcheckd.cfg(4)**, **logcheckd.pattern(4)** and **logcheckd.interface(3)** for more information.

The **GENERAL OPTIONS** options are loaded on daemon start, the **LOGFILE OPTIONS** options are loaded on daemon start and are re-evaluated in an interval of **300** seconds or when the **lgcheckd -a refresh** command is invoked.

The provided template file **var/samples/templates/lgcheck.cfg** contains the settings computed by the **lgcheckd** only and should be used as starting point for configuration of new setups if the legacy log checking daemon **logcheckd** was not used before.

(etc/logcheckd.cfg)

configuration file of **logcheckd**.

If the file **etc/lgcheckd.cfg** does not exist the **lgcheckd** reads this file without disadvantages. However, the **lgcheckd** computes the leaner set of options as documented in **lgcheckd.cfg(4)**.

lib/logcheckd/

This directory contains additional files used by **lgcheckd**.

lib/logcheckd/database/state.ddl

SQL data definition language (DDL) file to create the **state.db**.

lib/logcheckd/interface/

This directory contains the interfaces which allow **lgcheckd** to communicate to other tools.

lib/logcheckd/interface/<interfacefile>

Interface that communicates with other software.

See **logcheckd.interface(3)** for more information.

var/cache/logcheckd/

cache data for **lgcheckd**.

var/log/lgcheckd.log

logfile of **lgcheckd**.

var/logcheckd/pattern/

Location of the pattern files used for logfile analysis.

var/logcheckd/pattern/<patternfile>

Pattern file that can be used on multiple operating systems for logfile analysis.

The needed pattern files are loaded on on daemon start and if changed re-loaded in an interval of **300** seconds or when the **lgcheckd -a refresh** command is invoked.

See **logcheckd.pattern(3)** for more information.

var/logcheckd/pattern/<patternfile>.<OSID>

Pattern file for a specific operating system.

In the config file **etc/lgcheckd.cfg** the pattern file is specified *without* the **<OSID>** part. **lgcheckd** first checks if an operating system dependent pattern file exists and then uses the non-operating system dependent pattern file. This eases up configuration and enables to have identical configuration files across different operating systems.

See **logcheckd.pattern(3)** for more information.

var/logcheckd/pattern/lgcheckd

Pattern file to alert **lgcheckd** own events.

var/logcheckd/iconfig/

Location of the configuration files for the interfaces.

var/logcheckd/iconfig/<interfacefile>.cfg

Configuration file of the interfaces located in the **lib/logcheckd/interface/** directory.

See **logcheckd.interface(3)** for more information.

var/logcheckd/iconfig/lgcheckd.cfg

Configuration file of the **lgcheckd** interface used to send alert mails on events in the logfile **var/log/lgcheckd.log** of the **lgcheckd** only.

var/logcheckd/state/

Persistent state information of the **lgcheckd** daemon. All information herein is maintained by the **lgcheckd** command, do not edit it by hand.

var/logcheckd/state/state.db

database that contains all state information and collection data of the **lgcheckd**. The **state.db** is a SQLite (see: **sqlite(1)**) database.

var/logcheckd/report/

Default report output directory.

var/logcheckd/report/logcheck.<hostname>.<timestamp>.html

Report file generated invoking the **lgcheckd -a report** command.

The report is also suited for viewing with the **lynx(1)** browser.

var/samples/templates/lgcheck.cfg

Template configuration file for **lgcheckd**.

EXAMPLES

1) an example pattern file

In this example the logfile of **su(1)** command which allows to switch the user is checked. **su** writes successful user switches in the logfile as:

```
SU 06/05 11:03 + ta barney-root
```

unsuccessful attempts are logged as:

```
SU 06/05 11:03 - ta barney-root
```

The Log Check report in this example should exclude all successful switches of the *root* user to any user, due to the fact that we trust the *root* user:

```
[EXCLUDE]
```

```
. * + . * root-.*$
```

Non-successful tries to switch to the *root* user are classified to the *HIGH* section:

```
[HIGH]
```

```
. * - .*-root$
```

Successful switches to *root* are classified as *MEDIUM* because the user doing the switch knows the password and the assumption here is, that the user knows this password by purpose:

```
[MEDIUM]
```

```
. * + .*-root$
```

Non-successful switches to other users as *root* are also classified as *MEDIUM* due to the fact that the abuse of a *non-root* user has less impact as the abuse of the *root* user, but it also might be a brute force attack to get access to a user login:

```
. * -
```

All remaining entries are classified to the *VERIFY* section:

```
[VERIFY]
```

```
. *
```

The complete pattern file:

```
[ /root ]
[ root@acme001 ][-sh]: cat -n /opt/edrc/var/logcheckd/pattern/su

01 #
02 # logcheckd/pattern/su - logfile analysis pattern file for: su
03 #
04 # [00] 25.04.2008 CWa Initial Version
05 #
06
07 [EXCLUDE]
08
09 .* + .* root-.*$
10
11 [HIGH]
12
13 .* - .*-root$
14
15 [MEDIUM]
16
17 .* + .*-root$
18 .* -
19
20 [LOW]
21
22
23
24 [VERIFY]
25
26 .*
```

2) start the lgcheckd daemon

```
[ /root ]
[ root@acme001 ][-sh]: lgcheckd -a start
```

lgcheckd - a log file checking and reporting daemon, by Chr. Walther

lgcheckd-INFO: start daemon.

3) check lgcheckd status

```
[ /root ]
[ root@acme001 ][-sh]: lgcheckd -a status
```

lgcheckd - a log file checking and reporting daemon, by Chr. Walther

lgcheckd-INFO: daemon is not running, collection is 'enabled', interfaces

4) create a 'Log File Check' report

```
[ /root ]
[ root@acme001 ][-sh]: lgcheckd -a report

lgcheckd - a log file checking and reporting daemon, by Chr. Walther

lgcheckd-INFO: create report for collect run '13'.
lgcheckd-INFO: report file is '/opt/edrc/var/logcheckd/report/logcheck.ac
lgcheckd-INFO: collection match data for collect run '13' deleted.
lgcheckd-INFO: active collect run is now '14'.
done.
```

5) stop lgcheckd

```
[ /root ]
[ root@acme001 ][-sh]: lgcheckd -a stop

lgcheckd - a log file checking and reporting daemon, by Chr. Walther

lgcheckd-INFO: daemon not running.
```

6) reset lgcheckd

```
[ /root ]
[ root@acme001 ][-sh]: lgcheckd -a reset

lgcheckd - a log file checking and reporting daemon, by Chr. Walther

lgcheckd-INFO: daemon database initialized.
```

SEE ALSO

edrcintro(1), **osid(3)**, **crontab(1)**, **ecrontab(1)**, **edrcinit(1m)**, **egrep(1)**, **jobstart(1)**, **lgcheckd.cfg(4)**, **lgcheckd.state.db(4)**, **lgcpattern(3)**, **logcheckd.interface(3)**, **logcheckd.pattern(4)**, **logtail(1)**, **lynx(1)**, **sqlite(1)**, **tpl(1)**

NOTES

The **lgcheckd** daemon is a more efficient and leaner version of the **logcheckd(1m)** command. It is recommended to use **lgcheckd** for new setups

The **lgcheckd** can be started using **edrcinit(1m)**, for which the service handler **WA2Ledrc.lgcheckd** is available.

BUGS

-

AUTHOR

lgcheckd was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lgcheckd.cfg – configuration file for lgcheckd

SYNOPSIS

edrc/etc/lgcheckd.cfg
(**edrc/etc/logcheckd.cfg**)

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **lgcheckd** command.

The **lgcheckd** daemon can compute the configuration file **edrc/etc/logcheckd.cfg** of the **logcheckd (1m)** command for easy effortless migration (see also: **USE_LEGACY_COMMAND=False** in **logcheckd.cfg(4)**).

However, the **lgcheckd** command computes only the options as documented in this manual page.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The fileformat for the '**GENERAL OPTIONS**' is **OPTION=VALUE**

Between the **OPTION**, the **=** and the **VALUE** are no spaces.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a **VALUE**.

For the '**LOGFILES OPTIONS**', the fileformat is *field ; field ; field ; field ; field ; field ;* (ensure, that between the *field* and the separator **;** are no spaces).

OPTIONS**GENERAL OPTIONS****ALERT_MAIL_TO**

Recipient of the alert mail. Multiple recipients have to be specified as a comma separated list. If no **ALERT_MAIL_TO** is set, there is no alert mail sent on filesystem shortage and on resumption of log collection.

Example: **ALERT_MAIL_TO=fred.flintstone@acme.ch,support@acme.ch**

Default: **ALERT_MAIL_TO=""**

COLLECT

If this setting is set to *True* the log match data collection is enabled on startup and the matches are saved to the **state.db** for later analysis and reporting, if set to **False** log match data collection is disabled.

Be aware, that command line definitions using **lgcheckd -a [no]collect** has preference.

Example: COLLECT=False

Default: COLLECT=True

INTERFACESTART

If this setting is set to *True* the interfaces defined in the **INTERFACES=...** definition are started on positive match, when set to **False** the interfaces are not started on positive match.

Be aware, that command line definitions using **lgcheckd -a [no]ifstart** has preference.

Example: INTERFACESTART=False

Default: INTERFACESTART=True

CLEAN_COLLECT

If this setting is set to *True* the collected logfile match data is removed after report creation. However, the match count information is retained.

Example: CLEAN_COLLECT=False

Default: CLEAN_COLLECT=True

REPORTDIR

Directory where the 'Log File Check' reports are written to. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: REPORTDIR=/dat/report/ACME/LOGCHECK

Example: REPORTDIR=var/www/report/rpt/007/ACME/LOGCHECK

Default: REPORTDIR=var/logcheckd/report

INTERFACES

Interfaces to be executed when a certain check level matches. The setting is a space separated list of entries in the format: *<interface>:<LEVEL1>,<LEVEL2>*

Example: INTERFACES="mail:HIGH CAUnicenter:HIGH,MEDIUM"

Default: INTERFACES=""

SUSPENDTRIGGER

If the free space of the file system on which the **state.db** resides is lower then

`'size of state.db' * SUSPENDTRIGGER`

the collection of log file matches is suspended.

Example: SUSPENDTRIGGER=5

Default: SUSPENDTRIGGER=20

LOGFILES OPTIONS

List of logfiles to be checked for patterns.

It is not needed to restart the lgcheckd when logfile entries are added.

Pattern files can be added and modified during **lgcheckd** runtime.

In all other cases the **lgcheckd** needs to be restarted.

The format of the logfiles specification is:

resolve_mode ; patternfile ; interfaces ; res ; res ; logfile ;

(ensure, that between the *field* and the separator *;* are no spaces)

where the fields have the following content:

resolve_mode

the resolve mode needs to be set to **eval** or **dfft**.

However, it makes no difference in how the entries are computed, the *resolve_mode* field exists to be backward compatible to the configuration file for the **logcheckd(1m)** command.

patternfile pattern file. The pattern files are located in the **var/logcheckd/pattern** directory. If an operating system dependent pattern file with the name **<patternfile>.<OSID>** exists (e.g. **Cron.Solaris**), this file will be used for logfile analysis, else the pattern file as specified (e.g. **Cron**) will be used.

interfaces Interfaces to be executed when a certain check level matches for a certain logfile. The setting is a space separated list of entries in the format: **<interface>:<LEVEL1>,<LEVEL2>**. If the *interfaces* column is set for a logfile, the setting in the **INTERFACES** option is ignored for that logfile.

res reserved for future use, leave it empty.

logfile logfile to analyze.

In general the logfile has to be specified with an absolute path name.

However, if a relative path is specified, it is relative to the root of the WA2L/edrc installation (as returned by the **aproot(3)** command).

Example:

```
dflt;WA2Ledrc;;;var/log/edrc.log;
dflt;lgcheckd;;;var/log/lgcheckd.log;
dflt;All;;;etc/rc.log;
eval;MCSG;;;etc/cmcluster/*/*.cntl.log;
dflt;sudo;;;var/adm/sudo.log;
dflt;Samba;;;var/opt/samba/log.smbd;
dflt;syslog;;;var/adm/syslog/syslog.log;
```

FILES

var/samples/templates/lgcheck.cfg

Template configuration file for **lgcheckd**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **aproot(3)**, **csv(3)**, **osid(3)**, **lgcheckd(1m)**, **logcheckd.interface(3)**, **logcheckd.pattern(4)**, **logtail(1)**, **server_environment(3)**, **tpl(1)**

NOTES

-

BUGS

-

AUTHOR

lgcheckd.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

state.db – lgcheckd database state.db

SYNOPSIS

edrc/var/logcheckd/state/state.db

AVAILABILITY

WA2L/edrc

DESCRIPTION

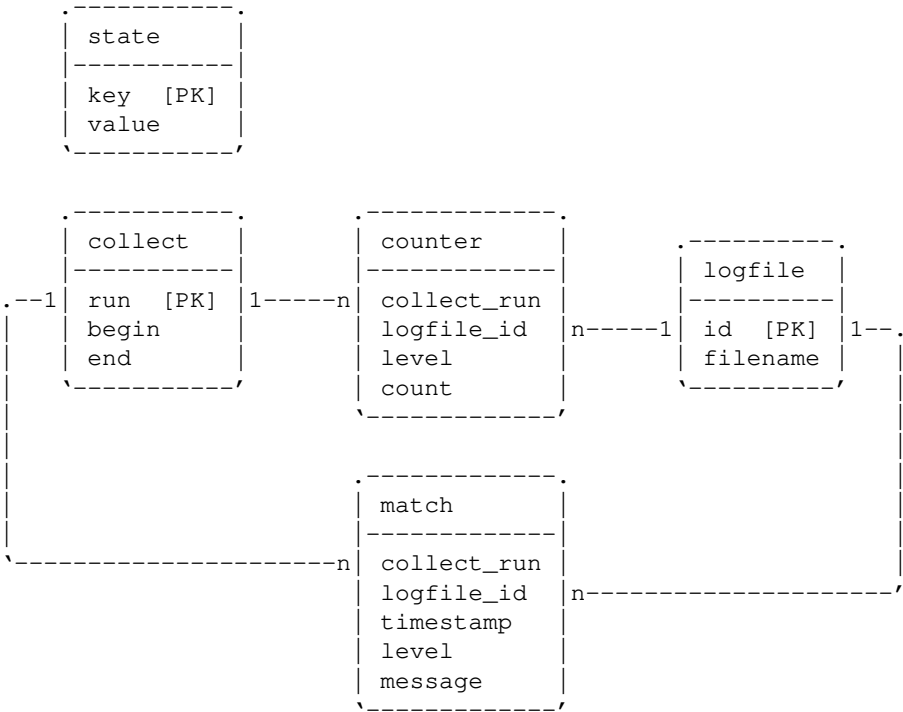
The **state.db** is the database that contains all volatile information of the log check daemon **lgcheckd**(1m).

FILEFORMAT

This is the ER (Entity Relationship) diagram of the **state.db** SQLite database that is used to store all log monitoring data of the log check daemon **lgcheckd**(1m).

The **[PK]** field is the primary key, **1---n** is a 'one to any' relationship.

All primary keys are unique in this data model.



| | |
|----------------|---|
| state | this table contains state information of the lgcheckd . |
| collect | <p>in this table the collect run properties are kept. Only one run is active, which is represented by an empty <i>end</i> field.</p> <p>Collect run entries are never deleted.</p> |
| logfile | the logfile table contains an index of all log files ever monitored. |
| counter | here all counts for matching patterns for each [LEVEL] are saved. Counters are never deleted. |
| match | <p>here the matching log file message data (except data of [EXCLUDE] sections and IGNORED messages) are saved. When a report is created (using -a report) the collected data of the related collection run is deleted by default.</p> <p>The data collection can be disabled using the COLLECT=False setting in the daemon configuration file or using the -a nocollect option on the command line.</p> <p>Data collection is automatically suspended when the available storage on the filesystem where the state.db resides gets tight. A suspended collection is automatically resumed after sufficient storage is available, again.</p> <p>If data collection is disabled or suspended the matches are still counted and saved to the counter table.</p> |

OPTIONS

-

FILES

| | |
|---|--|
| lib/logcheckd/database/state.ddl | SQL file to generate the SQLite database. |
| var/logcheckd/state/state.db | SQLite main database file. |
| var/logcheckd/state/state.db-wal | <p>this file is the Write-Ahead Log. It temporarily stores the changes made to the database. When a transaction is committed, the changes are appended to the wal file. Eventually, these changes are transferred (checkpointed) from the wal file back into the main database file.</p> |
| var/logcheckd/state/state.db-shm | <p>this file is the Shared Memory File. It is used to coordinate access between multiple database connections in WAL mode. The shared memory file is used to store the index of transactions in the wal file and to handle concurrency control.</p> |

EXAMPLES

-

SEE ALSO

edrcintro(1), **lgcheckd**(1m), **lgcheckd.cfg**(4), **logcheckd.cfg**(4), **logcheckd.pattern**(4), **sqlite**(1)

NOTES

-

BUGS

-

AUTHOR

state.db was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lgcpattern – check/verify a logcheckd and lgcheckd pattern file definition

SYNOPSIS

edrc/lib/lgcpattern [**-h**]

cat logfile | **lgcpattern** [**-s** | **-i**] *patternfile*

lgcpattern [**-s** | **-i**] *patternfile* < *logfile*

cat logfile | **lgcpattern** (**-p** | **-P**)

lgcpattern (**-p** | **-P**) < *logfile*

AVAILABILITY

WA2L/edrc

DESCRIPTION

utility to help to define a **logcheckd.pattern**(4) file.

OPTIONS

-h print usage message.

-s print match statistics for each pattern.

This output can be used to re-arrange the pattern sequence in the pattern file for best performance.

-i print only the **[IGNORED]** log lines, the lines that do not match to any pattern.

-p produce patterns from *logfile* stream to be used in *patternfiles*.

The output is reduced by a leading time stamp, characters that are special to regular expressions are escaped and numbers are replaced by related regular expressions.

-P as the **-p** option, but also print the commented (**#** ...) original log message line before the related pattern.

patternfile pattern file name as present in the pattern file directory.

Please note that the pattern file including an operating system id **osid**(3) has to be specified. This allows to also check pattern files of operating systems different to the one logged-on to be

checked.

logfile log file stream to check the *patternfile* against.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|------------------------------|
| 0 | no error. |
| 1 | pattern file does not exist. |
| 4 | usage message printed. |

FILES

edrc/var/logcheckd/pattern/
pattern file directory.

edrc/var/logcheckd/pattern/*patternfile*
pattern file to check/analyse/verify against a log file data stream.

EXAMPLES

1) Analyze parts of a log file

Use the last 10 lines of the **/var/log/syslog** log file and check it against the **syslog.Linux-64** pattern file.

This to see which message match against the *patternfile* and for which no matches exist (**=[IGNORED]**) and for which matches patterns are probably missing (**=[VERIFY]**).

```
tail -10 /var/log/syslog | \
lgcpattern syslog.Linux-64
```

2) Analyze pattern file definition

Use the last 10000 lines of the **/var/log/syslog** log file to verify the **syslog.Linux-64** pattern file.

This to see a statistics of message-match counts to patterns in the *patternfile*.

This result can be used to possibly rearrange the pattern sequence to improve analysis efficiency.

```
tail -10000 /var/log/syslog | \
lgcpattern -s syslog.Linux-64
```

3) Generate patterns based on a log file

Use the last 10 lines of the **/var/log/syslog** log file to generate patterns for the log lines.

Each "patternized" line is prepended by a commented original message line to ease up pattern definition.

```
tail -10 /var/log/syslog | \
lgcpattern -P
```

4) Generate unique patterns based on a log file

Use the last 10 lines of the **/var/log/syslog** log file to generate unique patterns for the log lines.

```
tail -10 /var/log/syslog | \
lgcpattern -p | sort -u
```

5) Generate patterns for un-categorized log entries

Use the last 500 lines of the **/var/log/syslog** log file to generate a list of unique patterns for the log lines that are currently not matched by any pattern of the pattern file **syslog.Linux-64**.

```
tail -500 /var/log/syslog | \
lgcpattern -i syslog.Linux-64 | \
lgcpattern -p | sort -u
```

SEE ALSO

edrcintro(1), **egrep(1)**, **lgcheckd(1m)**, **logcheckd(1m)**, **logcheckd.pattern(4)**, **osid(3)**, **regexintro(4)**

NOTES

-

BUGS

-

AUTHOR

lgcpattern was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

listtemp – list state of temporary directories in a directory

SYNOPSIS

edrc/lib/listtemp [-h]

listtemp -d *directory*

AVAILABILITY

WA2L/edrc

DESCRIPTION

list the state of all temporary directories, that are created in a certain directory.

The output is a list that provides the following information:

IS_ACTIVE

True = the process that created the directory is still running;

False = the process that created the directory is not running any more.

If an entry is marked as **False** the process that created the temporary directory is no longer running and the temporary directory can be removed without side effects.

DIRECTORY

the temporary directory.

CREATED

date and time when the temporary directory was created.

COMMAND

command name that created the directory.

PID

process id of the process that created the directory.

USER

name of the user that created the temporary directory.

PROCESS

process name of the process that created the directory.

OPTIONS

-h usage message.

-d *directory*
directory where to start to search for temporary directories.

ENVIRONMENT

-

EXIT STATUS

0 no error.

3 directory specified in the **-d** option does not exist.

4 usage message listed.

FILES

.maketemp.flagfile
flagfile located in the temporary directory containing the state information about the command that created the temporary directory. This file is created by **maketemp(3)** and is also read by **removetemp(3)** when removing the temporary directory.

EXAMPLES

-

SEE ALSO

edrcintro(1), **maketemp(3)**, **purgetemp(3)**, **removetemp(3)**

NOTES

-

BUGS

-

AUTHOR

listtemp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ll – list files (ls -laF)

SYNOPSIS

edrc/bin/ll [*ls_options*]... [*file*]...

AVAILABILITY

WA2L/edrc

DESCRIPTION

long listing of files.

Internally **ll** calls **ls -laF** to perform a long listing of the files.

OPTIONS

ls_options additional options to the native **ls**(1) command that is called with the **-laF** option.

file file(s) to be listed.

ENVIRONMENT

-

EXIT STATUS

x exist status of the native **ls**(1) command.

EXAMPLES

-

SEE ALSO

edrcintro(1), **ls**(1)

FILES

-

NOTES

-

BUGS

-

AUTHOR

It was developed by by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

llcomp – list the files contained in a compressed archive file

SYNOPSIS

edrc/bin/llcomp [**-h**]

llcomp *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **llcomp** you can list the files included in a compressed archive without decompressing the whole archive file. The output of **llcomp** is more detailed then the output of the **lscomp** command. Note, that the output of **llcomp** differs depending on the archive format, while the output of **lscomp** is identical for all supported archives.

OPTIONS

-h usage message

file filename of the compressed archive file. This file is a file with the suffixes **.tgz**, **.tar.gz**, **.tar.bz**, **.tar.bz2**, **.tar.Z**, **.tar.zip**, **.cpio**, **.cpio.gz**, **.cpio.bz**, **.cpio.bz2**, **.cpio.xz**, **.cpio.Z**, **.cpio.zip**, **.depot.gz**, **.depot**, **.epub**, **.zip**, **.jar**, **.whl**, **.rock**, **.rpm**, **.deb** or **.bw**.

EXIT STATUS

- 1** specified archive file not found.
- 2** file format of specified archive file not recognized. The file format is determined by the file suffix only.
- 4** usage displayed.
- 6** error while decompressing a compressed archive. This error occurs if the file is not compressed with the command that is normally assigned with the file suffix.
For instance: the file **my_archive.tar.gz** was not really compressed with **gzip** or the file is corrupt.
- 7** error while accessing file list in archive. This error occurs if the file is not archived with the command that is normally assigned to the file suffix.
For instance: the file **my_archive.tar.gz** was not really a compressed **tar** file or the tar archive

is corrupt.

SEE ALSO

edrcintro(1), **lscomp(1)**, **catcomp(1)**, **ar(1)**, **bwcreate(1)**, **bwcompare(1)**, **bzip2(1)**, **zip(1)**, **unzip(1)**, **compress(1)**, **cpio(1)**, **dpkg(1)**, **rpm(1)**, **jar(1)**, **tar(1)**, **gzip(1)**, **xz(1)**

BUGS

-

AUTHOR

llcomp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

locate – list files in databases that match a pattern

SYNOPSIS

```
locate [-d path | --database=path] [-e | -E | --[non-]existing] [-i | --ignore-case] [-0 | --null] [-c |
--count] [-w | --wholename] [-b | --basename] [-l N | --limit=N] [-S | --statistics] [-r | --regex ]
[--regextype R] [--max-database-age D] [-P | -H | --nofollow] [-L | --follow] [--version] [-A | --all]
[-p | --print] [--help] pattern...
```

DESCRIPTION

This manual page documents the GNU version of **locate**. For each given pattern, **locate** searches one or more databases of file names and displays the file names that contain the pattern. Patterns can contain shell-style metacharacters: ‘*’, ‘?’, and ‘[]’. The metacharacters do not treat ‘/’ or ‘.’ specially. Therefore, a pattern ‘foo*bar’ can match a file name that contains ‘foo3/bar’, and a pattern ‘*duck*’ can match a file name that contains ‘lake/.ducky’. Patterns that contain metacharacters should be quoted to protect them from expansion by the shell.

If a pattern is a plain string — it contains no metacharacters — **locate** displays all file names in the database that contain that string anywhere. If a pattern does contain metacharacters, **locate** only displays file names that match the pattern exactly. As a result, patterns that contain metacharacters should usually begin with a ‘*’, and will most often end with one as well. The exceptions are patterns that are intended to explicitly match the beginning or end of a file name.

The file name databases contain lists of files that were on the system when the databases were last updated. The system administrator can choose the file name of the default database, the frequency with which the databases are updated, and the directories for which they contain entries; see **updatedb(1)**.

If **locate**’s output is going to a terminal, unusual characters in the output are escaped in the same way as for the **--print** action of the **find** command. If the output is not going to a terminal, file names are printed exactly as-is.

OPTIONS

-0, --null

Use ASCII NUL as a separator, instead of newline.

-A, --all

Print only names which match all non-option arguments, not those matching one or more non-option arguments.

-b, --basename

Results are considered to match if the pattern specified matches the final component of the name of a file as listed in the database. This final component is usually referred to as the ‘base name’.

-c, --count

Instead of printing the matched filenames, just print the total number of matches we found, unless **--print** (**-p**) is also present.

-d path, --database=path

Instead of searching the default file name database, search the file name databases in *path*, which is a colon-separated list of database file names. You can also use the environment variable **LOCATE_PATH** to set the list of database files to search. The option overrides the environment variable if both are used. Empty elements in the path are taken to be synonyms for the file name of the default database. A database can be supplied on stdin, using ‘-’ as an element of *path*. If more than one element of *path* is ‘-’, later instances are ignored (and a warning message is printed).

The file name database format changed starting with GNU **find** and **locate** version 4.0 to allow machines with different byte orderings to share the databases. This version of **locate** can automatically recognize and read databases produced for older versions of GNU **locate** or Unix versions of **locate** or **find**. Support for the old locate database format will be discontinued in a future release.

- e, --existing*
Only print out such names that currently exist (instead of such names that existed when the database was created). Note that this may slow down the program a lot, if there are many matches in the database. If you are using this option within a program, please note that it is possible for the file to be deleted after **locate** has checked that it exists, but before you use it.
- E, --non-existing*
Only print out such names that currently do not exist (instead of such names that existed when the database was created). Note that this may slow down the program a lot, if there are many matches in the database.
- help* Print a summary of the options to **locate** and exit.
- i, --ignore-case*
Ignore case distinctions in both the pattern and the file names.
- l N, --limit=N*
Limit the number of matches to N. If a limit is set via this option, the number of results printed for the *-c* option will never be larger than this number.
- L, --follow*
If testing for the existence of files (with the *-e* or *-E* options), consider broken symbolic links to be non-existing. This is the default.
- max-database-age D*
Normally, **locate** will issue a warning message when it searches a database which is more than 8 days old. This option changes that value to something other than 8. The effect of specifying a negative value is undefined.
- m, --mmap*
Accepted but does nothing, for compatibility with BSD **locate**.
- P, -H, --nofollow*
If testing for the existence of files (with the *-e* or *-E* options), treat broken symbolic links as if they were existing files. The *-H* form of this option is provided purely for similarity with **find**; the use of *-P* is recommended over *-H*.
- p, --print*
Print search results when they normally would not, because of the presence of *--statistics* (*-S*) or *--count* (*-c*).
- r, --regex*
The pattern specified on the command line is understood to be a regular expression, as opposed to a glob pattern. The Regular expressions work in the same way as in **emacs** except for the fact that "." will match a newline. GNU **find** uses the same regular expressions. Filenames whose full paths match the specified regular expression are printed (or, in the case of the *-c* option, counted). If you wish to anchor your regular expression at the ends of the full path name, then as is usual with regular expressions, you should use the characters *^* and *\$* to signify this.
- regextype R*
Use regular expression dialect R. Supported dialects include 'findutils-default', 'posix-awk', 'posix-basic', 'posix-egrep', 'posix-extended', 'posix-minimal-basic', 'awk', 'ed', 'egrep', 'emacs', 'gnu-awk', 'grep' and 'sed'. See the Texinfo documentation for a detailed explanation of these dialects.
- s, --stdio*
Accepted but does nothing, for compatibility with BSD **locate**.
- S, --statistics*
Print various statistics about each locate database and then exit without performing a search, unless non-option arguments are given. For compatibility with BSD, *-S* is accepted as a synonym for *--statistics*. However, the output of **locate** *-S* is different for the GNU and BSD

implementations of **locate**.

--version

Print the version number of **locate** and exit.

-w, --wholename

Match against the whole name of the file as listed in the database. This is the default.

ENVIRONMENT

LOCATE_PATH

Colon-separated list of databases to search. If the value has a leading or trailing colon, or has two colons in a row, you may get results that vary between different versions of **locate**.

HISTORY

The **locate** program started life as the BSD fast find program, contributed to BSD by James A. Woods. This was described by his paper *Finding Files Fast* which was published in *Usenix ;login:*, Vol 8, No 1, February/March, 1983, pp. 8-10. When the **find** program began to assume a default **-print** action if no action was specified, this changed the interpretation of **find pattern**. The BSD developers therefore moved the fast find functionality into **locate**. The GNU implementation of **locate** appears to be derived from the same code.

Significant changes to **locate** in reverse order:

- 4.3.7 Byte-order independent support for old database format
- 4.3.3 **locate -i** supports multi-byte characters correctly
Introduced **--max_db_age**
- 4.3.2 Support for the **slocate** database format
- 4.2.22 Introduced the **--all** option
- 4.2.15 Introduced the **--regex** option
- 4.2.14 Introduced options **-L, -P, -H**
- 4.2.12 Empty items in **LOCATE_PATH** now indicate the default database
- 4.2.11 Introduced the **--statistics** option
- 4.2.4 Introduced **--count** and **--limit**
- 4.2.0 Glob characters cause matching against the whole file name
- 4.0 Introduced the **LOCATE02** database format
- 3.7 **Locate** can search multiple databases

BUGS

The **locate** database correctly handles filenames containing newlines, but only if the system's **sort** command has a working **-z** option. If you suspect that **locate** may need to return filenames containing newlines, consider using its **--null** option.

REPORTING BUGS

GNU findutils online help: <<https://www.gnu.org/software/findutils/#get-help>>

Report any translation bugs to <<https://translationproject.org/team/>>

Report any other issue via the form at the GNU Savannah bug tracker:

<<https://savannah.gnu.org/bugs/?group=findutils>>

General topics about the GNU findutils package are discussed at the *bug-findutils* mailing list:

<<https://lists.gnu.org/mailman/listinfo/bug-findutils>>

COPYRIGHT

Copyright © 1994-2022 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

find(1), **updatedb**(1m), **xargs**(1), **glob**(3), **locatedb**(4)

Full documentation <<https://www.gnu.org/software/findutils/locate>>

or available locally via: **info locate**

NAME

locatedb – front-compressed file name database

DESCRIPTION

This manual page documents the format of file name databases for the GNU version of **locate**. The file name databases contain lists of files that were in particular directory trees when the databases were last updated.

There can be multiple databases. Users can select which databases **locate** searches using an environment variable or command line option; see **locate(1)**. The system administrator can choose the file name of the default database, the frequency with which the databases are updated, and the directories for which they contain entries. Normally, file name databases are updated by running the **updatedb** program periodically, typically nightly; see **updatedb(1)**.

GNU LOCATE02 database format

This is the default format of databases produced by **updatedb**. The **updatedb** program runs **frcode** to compress the list of file names using front-compression, which reduces the database size by a factor of 4 to 5. Front-compression (also known as incremental encoding) works as follows.

The database entries are a sorted list (case-insensitively, for users' convenience). Since the list is sorted, each entry is likely to share a prefix (initial string) with the previous entry. Each database entry begins with a signed offset-differential count byte, which is the additional number of characters of prefix of the preceding entry to use beyond the number that the preceding entry is using of its predecessor. (The counts can be negative.) Following the count is a null-terminated ASCII remainder — the part of the name that follows the shared prefix.

If the offset-differential count is larger than can be stored in a signed byte (± 127), the byte has the value 0x80 (binary 10000000) and the actual count follows in a 2-byte word, with the high byte first (network byte order). This count can also be negative (the sign bit being in the first of the two bytes).

Every database begins with a dummy entry for a file called 'LOCATE02', which **locate** checks for to ensure that the database file has the correct format; it ignores the entry in doing the search.

Databases cannot be concatenated together, even if the first (dummy) entry is trimmed from all but the first database. This is because the offset-differential count in the first entry of the second and following databases will be wrong.

In the future, the data within the locate database may not be sorted in any particular order. To obtain sorted results, pipe the output of **locate** through **sort -f**.

slocate database format

The **slocate** program uses a database format similar to, but not quite the same as, GNU **locate**. The first byte of the database specifies its *security level*. If the security level is 0, **slocate** will read, match and print filenames on the basis of the information in the database only. However, if the security level byte is 1, **slocate** omits entries from its output if the invoking user is unable to access them. The second byte of the database is zero. The second byte is followed by the first database entry. The first entry in the database is not preceded by any differential count or dummy entry. Instead the differential count for the first item is assumed to be zero.

Starting with the second entry (if any) in the database, data is interpreted as for the GNU LOCATE02 format.

Old Locate Database format

There is also an old database format, used by Unix **locate** and **find** programs and earlier releases of the GNU ones. **updatedb** runs programs called **bigram** and **code** to produce old-format databases. The old format differs from the above description in the following ways. Instead of each entry starting with an offset-differential count byte and ending with a null, byte values from 0 through 28 indicate offset-differential counts from -14 through 14. The byte value indicating that a long offset-differential count follows is 0x1e (30), not 0x80. The long counts are stored in host byte order, which is not necessarily network byte order, and host integer word size, which is usually 4 bytes. They also represent a count 14 less than their value. The database lines have no termination byte; the start of the next line is indicated by its first byte having a

value ≤ 30 .

In addition, instead of starting with a dummy entry, the old database format starts with a 256 byte table containing the 128 most common bigrams in the file list. A bigram is a pair of adjacent bytes. Bytes in the database that have the high bit set are indexes (with the high bit cleared) into the bigram table. The bigram and offset-differential count coding makes these databases 20–25% smaller than the new format, but makes them not 8-bit clean. Any byte in a file name that is in the ranges used for the special codes is replaced in the database by a question mark, which not coincidentally is the shell wildcard to match a single character.

EXAMPLE

Input to **frcode**:

```
/usr/src
/usr/src/cmd/aardvark.c
/usr/src/cmd/armadillo.c
/usr/tmp/zoo
```

Length of the longest prefix of the preceding entry to share:

```
0 /usr/src
8 /cmd/aardvark.c
14 rmadillo.c
5 tmp/zoo
```

Output from **frcode**, with trailing nulls changed to newlines and count bytes made printable:

```
0 LOCATE02
0 /usr/src
8 /cmd/aardvark.c
6 rmadillo.c
-9 tmp/zoo
```

(6 = 14 - 8, and -9 = 5 - 14)

REPORTING BUGS

GNU findutils online help: <<https://www.gnu.org/software/findutils/#get-help>>

Report any translation bugs to <<https://translationproject.org/team/>>

Report any other issue via the form at the GNU Savannah bug tracker:

<<https://savannah.gnu.org/bugs/?group=findutils>>

General topics about the GNU findutils package are discussed at the *bug-findutils* mailing list:

<<https://lists.gnu.org/mailman/listinfo/bug-findutils>>

COPYRIGHT

Copyright © 1994-2022 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

find(1), **locate**(1), **xargs**(1), **locatedb**(4), **updatedb**(1m)

Full documentation <<https://www.gnu.org/software/findutils/locatedb>>
or available locally via: **info locatedb**

NAME
locations – print a list of company locations with office times

SYNOPSIS
edrc/bin/locations [-h]

AVAILABILITY
WA2L/edrc

DESCRIPTION
print a list of company locations with office times.

OPTIONS
-h usage message.

ENVIRONMENT
-

EXIT STATUS
4 usage printed.

0 always.

FILES
etc/locations.cfg
configuration file of **locations**.

EXAMPLES
Example output of the **locations** command:

| LOCAL-TIME | ON-DUTY | COUNTRY | COMPANY | CITY | COM |
|------------------|---------|---------|-----------|-------------|------|
| Fri Nov 06 09:47 | yes | Corea | PartnerRe | Seoul | Off. |
| Fri Nov 06 11:47 | yes | USA | EDS | Plano, TX | Hea |
| Fri Nov 06 12:47 | yes | Mexico | PartnerRe | Mexico City | Off. |
| Fri Nov 06 13:47 | yes | Canada | PartnerRe | Montreal | Off. |

locations(1)

General Commands

locations(1)

| | | | | | | |
|------------|-------|-----|-------------|-----------|---------------|------|
| Fri Nov 06 | 13:47 | yes | Canada | PartnerRe | Toronto | Off. |
| Fri Nov 06 | 13:47 | yes | USA | PartnerRe | Greenwich, CT | USA |
| Fri Nov 06 | 14:47 | yes | Argentina | PartnerRe | Buenos Aires | Off. |
| Fri Nov 06 | 14:47 | yes | Bermuda | PartnerRe | Hamilton | Hea |
| Fri Nov 06 | 14:47 | yes | Chile | PartnerRe | Santiago | Off. |
| Fri Nov 06 | 15:47 | yes | Brazil | PartnerRe | Buenos Aires | Off. |
| Fri Nov 06 | 18:47 | no | England | PartnerRe | Dublin | Off. |
| Fri Nov 06 | 18:47 | no | England | PartnerRe | London | Off. |
| Fri Nov 06 | 18:47 | no | Japan | PartnerRe | Tokyo | Off. |
| Fri Nov 06 | 18:47 | no | USA | PartnerRe | Tulsa | Off. |
| Fri Nov 06 | 19:47 | no | France | PartnerRe | Paris | COE |
| Fri Nov 06 | 19:47 | no | Germany | PartnerRe | Hannover | Off. |
| Fri Nov 06 | 19:47 | yes | Spain | EDS | Zaragoza | Mon. |
| Fri Nov 06 | 19:47 | no | Spain | EDS | Zaragoza | Pro |
| Fri Nov 06 | 19:47 | no | Switzerland | BEDAG | Bern | Ope |
| Fri Nov 06 | 19:47 | no | Switzerland | EDS | Beringen | Sys |
| Fri Nov 06 | 19:47 | no | Switzerland | EDS | Zuchwil | Sys |
| Fri Nov 06 | 19:47 | yes | Switzerland | EDS | Zuerich | 24 |
| Fri Nov 06 | 19:47 | no | Switzerland | EDS | Zuerich | Sys |
| Fri Nov 06 | 19:47 | no | Switzerland | PartnerRe | Zurich | COE |
| Fri Nov 06 | 19:47 | no | Switzerland | Tetrad | Guemligen | Win |
| Sat Nov 07 | 02:47 | no | China | PartnerRe | Hong Kong | Off. |
| Sat Nov 07 | 02:47 | no | Malaysia | PartnerRe | Malaysia | Off. |
| Sat Nov 07 | 02:47 | no | Singapore | PartnerRe | Singapore | Off. |

SEE ALSO

edrcintro(1), contrib.locations(1m), locations.cfg(4)

NOTES

-

BUGS

-

AUTHOR

locations was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

locations.cfg – locations configuration file

SYNOPSIS

etc/locations.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the location definition for the **locations** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

COUNTRY;TIMEZONE;COMPANY;CITY;COMMENTS;WORKTIME_FROM;WORKTIME_TO;

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

COUNTRY

Country of the (company) location.

TIMEZONE

Timezone where the location is in.

See **etc/timezone.dat** for a list of time zones.

COMPANY

Company or organization name.

CITY

City of the company or organization.

COMMENTS

Comments.

WORKTIME_FROM

time when the working time in the location starts. The time is the local time in the timezone as set in the *TIMEZONE* field.

WORKTIME_TO

time when the working time in the location ends. The time is the local time in the timezone as set in the *TIMEZONE* field.

The *WORKTIME_FROM* and *WORKTIME_TO* fields are used together with the *TIMEZONE* to resolve if a certain location is currently on duty.

EXAMPLES

```
#
# locations.cfg - configuration for locations
#
# [00] 08.03.2003 CWa Initial Version
#
USA:EST5EDT:PartnerRe:Greenwich, CT:USA-Branch:0800:1700:
Canada:EST5EDT:PartnerRe:Toronto:Office:0800:1700:
Canada:EST5EDT:PartnerRe:Montreal:Office:0800:1700:
Bermuda:AST4ADT:PartnerRe:Hamilton:Headquater:0800:1700:
Germany:MET-1METDST:PartnerRe:Hannover:Office:0800:1700:
Switzerland:MET-1METDST:PartnerRe:Zurich:COE Infrastructure:0800:1700:
France:MET-1METDST:PartnerRe:Paris:COE Finance:0800:1700:
Singapore:PST-8PDT:PartnerRe:Singapore:Office:0800:1700:
Argentina:EST4EDT:PartnerRe:Buenos Aires:Office:0800:1700:
China:GMT-8:PartnerRe:Hong Kong:Office:0800:1700:
Chile:GMT+4:PartnerRe:Santiago:Office:0800:1700:
Brazil:EBST3EBDT:PartnerRe:Buenos Aires:Office:0800:1700:
England:GMT:PartnerRe:Dublin:Office:0800:1700:
England:GMT:PartnerRe:London:Office:0800:1700:
Malaysia:MST-8:PartnerRe:Malaysia:Office:0800:1700:
Mexico:CST6CDT:PartnerRe:Mexico City:Office:0800:1700:
Corea:GMT+9:PartnerRe:Seoul:Office:0800:1700:
Japan:JST:PartnerRe:Tokyo:Office:0800:1700:
USA:CST:PartnerRe:Tulsa:Office:0800:1700:
USA:MST7MDT:EDS:Plano, TX:Headquater:0800:1700:
Switzerland:MET-1METDST:Tetrad:Guemligen:Windows-Experts:0800:1700:
Spain:MET-1METDST:EDS:Zaragoza:Monitoring UX:0000:2400:
```

SEE ALSO

edrcintro(1), timezone(3), etc/timezone.dat, /usr/lib/tztab, <http://www.timezoneconverter.com>

NOTES

-

BUGS

-

AUTHOR

locations.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

log – write a message to a logfile

SYNOPSIS

edrc/lib/log *level* "*message text*"

log -h

AVAILABILITY

WA2L/edrc

DESCRIPTION

This command is used in scripts to write a standardized message to a log file.

The message is appended to a logfile and has the format:

YYYY-MM-DD HH:MM:SS [<Session>] <Scriptname>-<LEVEL>: <message text>

Example:

2008-12-14 14:00:05 [SAT_6031] my_script-ERROR: dir '/dat/report/myreport' does not exist.

The **<Scriptname>** is derived from the **\$EDRC_SCRIPTNAME** environment variable, the **<LEVEL>** is the upper case of the *level* string given as the first option of the command. The **<Session>** is derived from the **\$EDRC_SESSION** environment variable if set, else the process id of the calling script is logged. The message text is written as received from the other options of the command.

OPTIONS

level

Message level. The string given here is printed in upper case. There is no restriction in the *level* strings, a convention used in commands within WA2L/edrc is:

INFO Information messages.

ERROR

An error occurred, that questions the correct output or function of the command. An error has to be corrected to ensure that the command can complete successfully.

WARNING

A non critical malfunction of the command occurred. Often a **WARNING** can be ignored, but in some occasions a warning can also lead to a minor reduction in output quality. A warning has to be analyzed and rated by the user.

FATAL

A fatal error is a condition that has to be analyzed and that needs user intervention and correction to ensure proper functionality.

message text

message to be printed.

ENVIRONMENT**\$EDRC_SCRIPTNAME**

scriptname that is printed in the **<Scriptname>** part of the output.

\$EDRC_LOGFILE

logfile where the log entries are appended to.

\$TZ

time zone.

EXIT STATUS

0 no error.

4 usage listed.

EXAMPLES

The following examples are script cut-outs of Bourne-, Korn- or Bash shell scripts:

1) common usage

```
EDRC_SCRIPTNAME=`basename $0`; export EDRC_SCRIPTNAME
EDRC_LOGFILE=/var/log/`basename $0`.log; export EDRC_LOGFILE

log INFO "command started"
```

2) set a session usage

```
EDRC_SCRIPTNAME=`basename $0`; export EDRC_SCRIPTNAME
EDRC_LOGFILE=/var/log/`basename $0`.log; export EDRC_LOGFILE
EDRC_SESSION=SAT_6031; export EDRC_SESSION

log INFO "command started"
```

SEE ALSO

edrcintro(1), msg(3), sh(1), ksh(1)

NOTES

-

BUGS

-

AUTHOR

log was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcat – cat remote- or local (time-stamped) log file(s)

SYNOPSIS

edrc/bin/logcat [**-h** | **-p**]

logcat *file* ...

logcat [[*user@*]*host*:]*file* ...

logcat [*options*] [**-f** "*from_ts*"] [**-t** "*to_ts*"] [[*user@*]*host*:]*file* ...

logcat [*options*] [**-f** "*from_ts*"] [**-t** "*to_ts*" **-t** "*to_ts*"] [[*user@*]*host*:]*file* ...

lcat ...

options ::= [**-l** *localuser*] [**-M**] [**-s** *num*] [**-NUM**] [**-T** *id*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

cat selected time range (from/to) in log *files*.

If the log file has time stamps, the data between a *from* and *to* time stamp is selected and **cated**.

Hint: To **grep** in log files with time stamps, use the **loggrep**(1) command directly instead of piping the output of **logcat**(1) to **grep** or **egrep**.

When there are no time stamps in the log file, or the time stamp format is not recognized, the whole log file data is **cated**.

The first **97** lines of the log file are used to resolve the time stamp format used in a particular file.

Beside text log file(s), **logcat** can handle file(s) which are **zipped** (**.zip**), **compressed** (**.Z**), **gzipped** (**.gz**), **bzip2ed** (**.bz**, **.bz2**, **.bzip2**) or **xzed** (**.xz**) directly.

To increase performance on scanning large log files which are bigger then **100 MByte**, only the last **1'000'000** lines are analysed. To force to nevertheless analyse the whole file, specify **-0** in the **-NUM** option.

FROM/TO DATA SELECTION

The **-f** "*from_ts*" and **-t** "*to_ts*" time stamps have to be specified in military format ("**2021-01-18 20:57:10**").

The following abbreviations and defaults apply (examples when executed on **Monday, 2021-01-18**):

| specified from/to_ts | resulting from_ts | resulting to_ts |
|----------------------|---------------------|---------------------|
| | 2021-01-18 00:00:00 | 2021-01-18 23:59:59 |
| 13:45 | 2021-01-18 13:45:00 | 2021-01-18 13:45:59 |
| 13:45:55 | 2021-01-18 13:45:55 | 2021-01-18 13:45:55 |
| 16 | 2021-01-16 00:00:00 | 2021-01-16 23:59:59 |
| 01-16 | 2021-01-16 00:00:00 | 2021-01-16 23:59:59 |
| 05-30 | 2020-05-30 00:00:00 | 2020-05-30 23:59:59 |
| 2020 | 2020-01-01 00:00:00 | 2020-12-31 23:59:59 |
| 2020-03 | 2020-03-01 00:00:00 | 2020-03-31 23:59:59 |
| 2020-03-27 | 2020-03-27 00:00:00 | 2020-03-27 23:59:59 |
| 2020-03-27 13 | 2020-03-27 13:00:00 | 2020-03-27 13:59:59 |
| 2020-03-27 13:45 | 2020-03-27 13:45:00 | 2020-03-27 13:45:59 |
| 2020-03-27 13:45:12 | 2020-03-27 13:45:12 | 2021-03-27 13:45:12 |
| yesterday | 2021-01-17 00:00:00 | 2021-01-17 23:59:59 |
| mon | 2021-01-11 00:00:00 | 2021-01-11 23:59:59 |
| tue | 2021-01-12 00:00:00 | 2021-01-12 23:59:59 |
| wed | 2021-01-13 00:00:00 | 2021-01-13 23:59:59 |
| thu | 2021-01-14 00:00:00 | 2021-01-14 23:59:59 |
| fri | 2021-01-15 00:00:00 | 2021-01-15 23:59:59 |
| sat | 2021-01-16 00:00:00 | 2021-01-16 23:59:59 |
| sun | 2021-01-17 00:00:00 | 2021-01-17 23:59:59 |

LOG FILE TIME STAMP FORMATS

The following time stamps are recognized (per log file):

- 1) 2021-01-30 23:59:59
- 2) 2021/01/30-23:59:59
- 3) Jan 30 23:59:59
- 4) ---- 2021-01-30 23:59:59 ----
- 5) 12/16-11:34:21
- 6) Nov 18, 2019 9:44:43 PM
Apr 15, 2020 10:50:51 AM
- 7) End-Date: 2021-01-15 10:06:29
Start-Date: 2021-01-15 10:06:35
- 8) [18/Jan/2021:10:51:23 +0100]
- 9) 2021/01/30 23:59:59
- 10) ----BEGIN: Mon Nov 18 19:51:51 UTC 2019
----END: Mon Nov 18 19:51:52 UTC 2019

- 11) 2020-10-27T17:44:21
- 12) #1612518675
- 13) 2021-01-30-23:59:59
- 14) [Tue Aug 27 10:56:41 2019]
[Tue Aug 27 10:56:41.1234 2019]
- 15) Thu Feb 4 12:27:12 2021
- 16) -80238 | Fri Feb 5 12:27:11 2021
447752 | Fri Feb 5 12:27:12 2021
- 17) update-alternatives 2020-10-27 22:50:52
- 18) 0156 01/31/21 12:23:56
0157 01/31/21 12:24:07
- 19) ---- Rules Applied on Monday 2021-01-11 12:25:48
- 20) [UTC 01/31/21 11:22:32]
- 21) Log started: 2021-01-15 10:06:29
Log ended: 2021-01-15 10:06:35
- 22) Log time: 2018-08-18 21:10:09
- 23) error 2021-02-13 06:12:51
info 2021-02-13 06:12:51
- 24) ERROR: apport (pid 593) Fri Feb 1 23:09:19 2019

OPTIONS

- h** usage message.
- p** print the timestamp formats known by **logcat**, some example files where that timestamp format has been discovered and the regular expression that is used to identify the timestamp.
- l *localuser***
use the local user *localuser* to initiate the connection.
- s *num*** use the first *num* lines of the logfile for timestamp format detection.

-NUM analyse the last *NUM* lines, instead of the whole file.

On large log files that exceed **100 MByte**, only the last **1'000'000** rows are scanned, whereas the first **4007** lines are here used to resolve the time stamp format.

To force to scan nevertheless the whole log file, specify **-0** here.

-f "from_ts"

start of data selection in military (=ISO) format, an abbreviated date or time specification as listed in the table in section **DESCRIPTION** or a past weekday (as: **yesterday, mon, tue, wed, thu, fri, sat, sun**).

Default is the current date at **00:00:00** (example: "**2021-01-18 00:00:00**").

-t "to_ts"

end of data selection in military (=ISO) format, an abbreviated date or time specification as listed in the table in section **DESCRIPTION** or a past weekday (as: **yesterday, mon, tue, wed, thu, fri, sat, sun**).

Default is current date at **23:59:59** (example: "**2021-01-18 23:59:59**").

-M

print meta data of **logcat** results to file descriptor **3**.

Example:

```
LOGGREGP_FROM="2020-03-27 13:45:00"
LOGGREGP_TO="2020-03-27 18:00:00"
LOGGREGP_COUNT="68312"
LOGGREGP_BEGIN="2020-03-27 13:50:18"
LOGGREGP_END="2020-03-27 17:58:36"
LOGGREGP_DURATION="000:04:13:24"
```

This output can be used to set variables in a script:

```
#!/bin/sh

logcat -M -f "2020-03-27 13:45:00" -t "2020-03-27 18:00:00" \
    starting ~edrc/var/log/edrc.log 3>/tmp/meta.out

. /tmp/meta.out
echo "ENTRIES: $LOGGREGP_COUNT between $LOGGREGP_BEGIN and $LOGGREGP_END"
```

[[user@]host:]file

remote- or local file.

If a **-** is specified as a *file* option, data is read from **stdin**.

Beside text files, the specified *file* can also be **zipped** (**.zip**), **compressed** (**.Z**), **gzipped** (**.gz**), **bzip2ed** (**.bz**, **.bz2**, **.bzip2**) or **xzed** (**.xz**).

ENVIRONMENT**\$LOGGREG_FROM**

from_ts as specified on the command line. However, the command line option **-f** "*from_ts*" has preference.

\$LOGGREG_TO

to_ts as specified on the command line. However, the command line option **-t** "*to_ts*" has preference.

EXIT STATUS

0

2 error occurred.

4 usage printed.

5 command has been aborted.

11 could not claim a temporary directory in **/tmp/**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **cat(1)**, **edrcsetup(1m)**, **egrep(1)**, **grep(1)**, **loggrep(1)**, **rcat(1)**, **regexintro(4)**, **ssh-exec(1)**, **ssh-exec.cfg(4)**

NOTES

logcat uses **rcat(1)** internally to get the remote- and local files.

A call to **lcat** ... is identical to a call to the **logcat** ... command.

logcat is a simplified use (or shortcut) to the **loggrep** command to display the whole section between two timestamps without **grep**ing the log contents. That's why some (environment) variables also for **logcat** have the **LOGGREG_...** prefix.

BUGS

-

AUTHOR

logcat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcheckd – daemon to check, classify and report logfile content

SYNOPSIS

edrc/bin/logcheckd [**-h**]

logcheckd -a (**start** | **stop** | **report** | **reset** | **status** | **defined** | **monitored**)

cat logfile | **logcheckd -a verify patternfile check_levels**

AVAILABILITY

WA2L/edrc

DESCRIPTION

NOTE: The **lgcheckd** daemon is a more efficient and leaner version of the **logcheckd(1m)** command. It is recommended to use **lgcheckd(1m)** at least for new setups.

With **logcheckd** logfiles can be analyzed and the content can be classified to different levels of importance.

During analysis the timestamp when a certain log entry has been discovered is added to the collected analysis output.

The levels used to classify the logfile content can be defined in pattern files which are related to one or more logfiles to be analyzed.

The generated HTML report allows easy navigation in the report and provides an index of the number of selected rows (=hits) per level of importance in an overview section of the report.

To allow judgement of the number of hits the absolute number of hits are also normalized to 'hits per day' (hits/day) which then are less dependent of the reporting period and can be compared to the hits/day of the last analysis period and a calculated hits trend.

The trend is calculated as:

$$\text{trend_new} = (0.75 * \text{trend_last}) + (0.25 * \text{hits_last})$$

This means that the past goes into the trend calculation with three quarters and the last run goes into the trend with a weight of one quarter. This causes to approach a good prediction of expected future logfile content hits.

The generated report can be kept on the file system, but the main purpose is to send it per mail to system administration personnel which will get a good overview of the log situation on the system.

For compliance the report contains a random compliance code that is used to prove that the report has been viewed. To help the administrative personnel and to enable a consistent compliance procedure the generated report provides a '**send report view confirmation mail**' hyperlink that prepares a compliance mail when clicking on it having the compliance code in the subject line. So in the mailbox it is easy to see which

reports have been viewed by sorting it by subject.

logcheckd will not fill up the file system where the log data is collected and reported. If file space will be short during log file analysis the analysis is suspended until sufficient disk space is available again and will then resume automatically. A report will also only be generated if sufficient free space is available. The following suspension criteria applies:

$$\{1.5 * lsize\} > \{available - (available + used) / 100 * MIN_FREESPACE\}$$

For a logfile analysis, the "lsize" is the total size of the logfile to be analyzed; for a report creation, the "lsize" is the size of all collected matches of the collect run for which the report should be generated.

logcheckd is also able to communicate to other commands for each matched logfile entry via the **INTERFACES** mechanism, which can be configured in the **logcheckd.cfg** configuration file. Using interfaces, it is therefore possible to forward discovered matches to a monitoring system that might not have a very sophisticated logfile analysis functionality.

REPORT STRUCTURE

The generated report is structured into the following sections:

General General information about the report. This section contains the following information:

Customer

Name of the Customer.

Environment

Name and description of the environment of which the host is part of.

Hostname

Hostname where the report has been generated.

Report Name of the report.

Report ID

Unique ID of the report.

Report Period

Period (date and time from, date and time to) that is covered by the report. To allow easy calculations the duration in seconds is provided, too.

Date Date when the report has been generated.

Generated by

Path and name of the command and user that generated (and mailed) the report.

Confirmation Code

Random confirmation code for compliance usage.

Compliance

Hyperlink providing an email with a structured subject and content that can be mailed to a compliance recipient.

Overview Overview of all logfiles that existed during the 'Report Period'. This section contains the following information:

Current (hits absolute)

Absolute number of hits per defined level and absolute number of ignored logfile rows during the 'Reporting Period'.

Current (hits/day)

Number of hits per defined level and number of ignored logfile rows during the 'Reporting Period' converted into hits per day.

Last (hits/day)

Number of hits per defined level and number of ignored logfile rows during the last 'Reporting Period' converted into hits per day.

Trend (hits/day)

Trend of hits per defined level and number of ignored logfile rows calculated based on all past reports.

Logfile Name and size of the logfile.

Logfiles All captured logfile rows sorted by logfile and defined level.

Footline Information about the **logcheckd** version used to generate the report.

OPTIONS

-h usage message.

-a action:

start

start the log check daemon **logcheckd** in the background. It is possible to start one daemon per system.

stop

stop a running **logcheckd** daemon.

report

create a LogCheck report for the period since the last report generation. If an email address is defined in the **REPORT_MAIL_TO** setting in the config file **etc/logcheckd.cfg** the report will be mailed to the recipient(s) defined in **REPORT_MAIL_TO**, **REPORT_MAIL_CC** and **REPORT_MAIL_BCC**.

reset

initialize the **logcheckd** daemon. All state information, collected logfile content hits, all counters and trend information will be cleared. If you configured to keep the logfile collection hits (**CLEAN_COLLECT=False**) all collect runs are removed. If you configured to keep the generated reports after mailing them (**CLEAN_REPORT=False**) the reports will be moved to the **REPORT-DIR/\$TODAY** directory.

status check if a **logcheckd** is running (EXIT STATUS: 14) or not running (EXIT STATUS: 15) on the system.

defined

list the logfiles defined in the **logcheckd.cfg** config file.

monitored

list the currently monitored logfiles including some additional information.

verify *patternfile check_levels*

verify the *check_levels* of the *patternfile* for the *logfile* piped into **logcheckd**. This option can be used to check if the settings made in a pattern file lead into the expected results. *check_levels* is a comma separated list of levels to be checked. This is equal to the **CHECK_LEVELS** setting in the **logcheckd.cfg** configuration file. When verifying a *patternfile* no timing information is added to the output.

ENVIRONMENT

-

SIGNALS

The following signals are handled by **logcheckd**. Do not use other signals as those listed below, as long as you do not really know what you are doing and what the consequences are. In general there is no need to invoke those signals by your own, this signals are used for inter process communication of **logcheckd**.

TERM kill the process without an argument (**kill PID**). Issuing this signal, the daemon will end.

EXIT STATUS

- 0** no error.
- 1** configfile **etc/logcheckd.cfg** does not exist.
- 2** operating system is not supported. See **osid(3)** if you get this error.
- 3** pattern file does not exist. This exit status is returned only when calling **logcheckd** with the **-a verify** option.

- 4 usage listed.
- 5 **logcheckd** has been aborted issuing <Ctrl>+<C>. This applies only when using the options **-a stop**, **-a reset** or **-a report**.
- 6 cannot write to **REPORTDIR**.
- 7 cannot write to **COLLECTDIR**.
- 8 cannot write to the **logcheckd** logfile.
- 9 cannot write to **LOCKDIR**.
- 10 directory **PATTERNDIR** does not exist.
- 11 a temporary directory could not be claimed or created in **/var/tmp**. Check the system temporary directory **/var/tmp** if you get this error, it is an indicator of system intrusion.
- 12 ambiguous pattern file name. This error only occurs if **logcheckd** is called with the **-a verify** option. During daemon operation logfile specifications containing ambiguous pattern file names are ignored and the daemon does not exit.
- 13 report creation failed.
- 14 daemon running.
- 15 daemon not running.
- 16 directory **INTERFACECONFIGDIR** does not exist.
- 17 not all interfaces, as specified in **INTERFACES** in the **logcheckd.cfg** config file, can be accessed. An interface has to be executable for the root user.

FILES

etc/logcheckd.cfg

configuration file of **logcheckd**, see **logcheckd.cfg(4)**, **logcheckd.pattern(4)** and **logcheckd.style(4)** for more information. The logfile options are reloaded in the interval defined in **CHECK_INTERVAL**, therefore logfiles can be added, modified and removed from the list without restarting the **logcheckd** daemon. When a logfile is removed, it will not be checked from that moment on, but it will be included in the first report generated afterwards.

lib/logcheckd/

This directory contains additional files used by **logcheckd**.

lib/logcheckd/style/

This directory contains the stylesheets used by the **logcheckd** when a HTML report is generated.

lib/logcheckd/style/WA2L.css

Default style sheet when generating the report.

lib/logcheckd/interface/

This directory contains the interfaces which allow **logcheckd** to communicate to other tools.

lib/logcheckd/interface/<interfacefile>

Interface that communicates with other software. See **logcheckd.interface(3)** for more information.

var/log/logcheckd.log

logfile of **logcheckd**. The location of the logfile can be configured in the **etc/logcheckd.cfg** config file.

var/logcheckd/pattern/

Location of the pattern files used for logfile analysis.

var/logcheckd/pattern/<patternfile>

Pattern file that can be used on multiple operating systems for logfile analysis.

var/logcheckd/pattern/<patternfile>.<OSID>

Pattern file for a specific operating system.

In the config file **etc/logcheckd.cfg** the pattern file is specified *without* the **<OSID>** part. **logcheckd** first checks if a operating system dependent pattern file exists and then uses the non-operating system dependent pattern file. This eases up configuration and enables to have identical configuration files across different operating systems.

var/logcheckd/iconfig/

Location of the configuration files for the interfaces.

var/logcheckd/iconfig/<interfacefile>.cfg

Configuration file of the interfaces located in the **lib/logcheckd/interface/** directory. See **logcheckd.interface(3)** for more information.

var/logcheckd/collect/

Default collect directory.

var/logcheckd/collect/<collect_run>/

This directory contains the collected logfile entries (hits) of all checked logfiles for a certain collect run.

var/logcheckd/collect/<collect_run>/<logfile_number>.excluded

Number of excluded ([EXCLUDE] section) rows of a checked logfile.

var/logcheckd/collect/<collect_run>/<logfile_number>.ignored

Number of ignored rows of a checked logfile.

var/logcheckd/collect/<collect_run>/<logfile_number>.<LEVEL>

Selected rows of a logfile for a certain *LEVEL* as configured in the pattern file. See also **logcheckd.pattern(4)** for more information about pattern files.

var/logcheckd/collect/<collect_run>/ts.begin

Begin timestamp of a logfile analysis / reporting period.

var/logcheckd/collect/<collect_run>/ts.end

End timestamp of a logfile reporting period..

var/logcheckd/state/

Persistent state information of the **logcheckd** daemon. All information herein is maintained by the **logcheckd** command, do not edit it by hand.

var/logcheckd/state/index

Index of all configured logfiles that have been found on the system.

var/logcheckd/state/collect.seq

Counter of collect runs.

var/logcheckd/state/report.seq

Counter of generated reports.

var/logcheckd/state/files/

File properties that have to be saved across collect runs.

var/logcheckd/state/files/<logfile_number>.inode

Inode number of a checked logfile.

var/logcheckd/state/files/<logfile_number>.rows

Number of rows of a checked logfile.

var/logcheckd/state/files/<logfile_number>.size

Size in kBytes (1 kByte = 1024 Bytes) of a checked logfile.

var/logcheckd/state/files/<logfile_number>.<LEVEL>.hits.last

Number of hits per day of a logfile for a certain *LEVEL*.

var/logcheckd/state/files/<logfile_number>.<LEVEL>.hits.trend

Trend of hits per day of a logfile for a certain *LEVEL*.

var/logcheckd/state/files/<logfile_number>.ignored.last

Number of ignored logfile rows per day of the last reporting period.

var/logcheckd/state/files/<logfile_number>.ignored.trend

Trend of number of ignored rows per day.

var/logcheckd/state/files/<logfile_number>.excluded.last

Number of excluded logfile rows per day of the last reporting period.

var/logcheckd/state/files/<logfile_number>.excluded.trend

Trend of number of excluded rows per day.

var/logcheckd/report/

Default report output directory.

<REPORTDIR>/<customer>_<report>_<environment_name>_<report_id>.html

Report file if it is saved.

var/samples/templates/logcheck.cfg

Template configuration file for **logcheckd**.

EXAMPLES

1) verify a pattern file

In this example the logfile of **su(1)** command which allows to switch the user is checked. **su** writes successful user switches in the logfile as:

```
SU 06/05 11:03 + ta barney-root
```

unsuccessful attempts are logged as:

```
SU 06/05 11:03 - ta barney-root
```

The LogCheck report in this example should exclude all successful switches of the *root* user to any user, due to the fact that we trust the *root* user:

```
[EXCLUDE]
```

```
. * + . * root-. * $
```

Non-successful tries to switch to the *root* user are classified to the *HIGH* section:

```
[HIGH]
```

```
. * - . *-root $
```

Successful switches to *root* are classified as *MEDIUM* because the user doing the switch knows the password and the assumption here is, that the user knows this password by purpose:

```
[MEDIUM]
```

```
. * + . *-root $
```

Non-successful switches to other users as *root* are also classified as *MEDIUM* due to the fact that the abuse of a *non-root* user has less impact as the abuse of the *root* user, but it also might be a brute force attack to get access to a user login:

```
. * -
```

All remaining entries are classified to the *VERIFY* section:

```
[VERIFY]
```

```
. *
```

Later when running **logcheckd** as daemon the *VERIFY* and *EXCLUDE* sections will only be collected/reported when they are listed in the **CHECK_LEVELS** setting (e.g: **CHECK_LEVELS=HIGH,MEDIUM,LOW,VERIFY,EXCLUDE**) in the config file **logcheckd.cfg**(4).

```
[ /root ]
[ root@acme001 ] [-sh]: cat -n /opt/edrc/var/logcheckd/pattern/su

01 #
02 # logcheckd/pattern/su - logfile analysis pattern file for: su
03 #
04 # [00] 25.04.2008 CWa Initial Version
05 #
06
07 [EXCLUDE]
08
09 . * + . * root-. *$
10
11 [HIGH]
12
13 . * - . *-root$
14
15 [MEDIUM]
16
17 . * + . *-root$
18 . * -
19
20 [LOW]
21
22
23
24 [VERIFY]
25
26 . *
```

```
[ /root ]
[ root@acme001 ] [-sh]: tail -16 /var/adm/sulog | \
    ~edrc/bin/logcheckd -a verify su EXCLUDE,HIGH,MEDIUM,LOW,VERIFY

logcheckd - a log file checking and reporting daemon, by Chr. Walth
verify patternfile '/opt/edrc/var/logcheckd/pattern/su' ...
```

```

[EXCLUDE]
009| SU 06/05 11:00 + tty?? root-oracle
009| SU 06/05 11:01 + tty?? root-edrc
009| SU 06/05 11:01 + tty?? root-edrc
009| SU 06/05 12:00 + tty?? root-oracle
009| SU 06/05 12:00 + tty?? root-oracle
009| SU 06/05 12:00 + tty?? root-oracle
009| SU 06/05 12:00 + tty?? root-oracle
009| SU 06/05 12:02 + tty?? root-edrc
009| SU 06/05 12:02 + tty?? root-edrc
009| SU 06/05 12:02 + tty?? root-edrc
(10)
[HIGH]
013| SU 06/05 11:03 - ta barney-root
(1)
[MEDIUM]
018| SU 06/05 11:02 - ta barney-fred
017| SU 06/05 11:03 + ta barney-root
017| SU 06/05 11:32 + tc barney-root
017| SU 06/05 11:33 + 2 fred-root
(4)
[LOW]
(0)
[VERIFY]
026| SU 06/05 11:40 + ta barney-fred
(1)
ignored
(0)
done.

```

2) start logcheckd

```

[ /root ]
[ root@acme001 ][-sh]: logcheckd -a start

logcheckd - a log file checking and reporting daemon, by Chr. Walther

start logcheckd daemon ...(PID=23757)... done.

```

3) check logcheckd status

```

[ /root ]
[ root@acme001 ][-sh]: logcheckd -a status

logcheckd - a log file checking and reporting daemon, by Chr. Walther

logcheckd daemon status ...(running,PID=23757,revision=33,collectrun=5)..

```

4) create a LogCheck report

```

[ /root ]
[ root@acme001 ][-sh]: logcheckd -a report

```



```

logcheckd - a log file checking and reporting daemon, by Chr. Walther

create report ...
  report information ...
    Customer ..... : WA2L
    Report ID ..... : acme001_10000
    Report Period .. : 2008-06-25 15:20:00 - 2008-06-29 00:39:48
    Send to .....  : admin@acme.ch
    Collect Run .... : 5
  done.
  write report ...
    header section ... done.
    general section ... done.
    overview section ... done.
    logfile section ... done.
    foot section ... done.
  done.
  send report ... done.
  clear collect directory ... done.
  clear report ... done.
done.

```

5) stop logcheckd

```

[ /root ]
[ root@acme001 ][-sh]: logcheckd -a stop

logcheckd - a log file checking and reporting daemon, by Chr. Walther

stop logcheckd daemon ...
  send logcheckd daemon the stop signal ... done.
  wait for daemon to stop .... done.
done.

```

6) reset logcheckd

```

[ /root ]
[ root@acme001 ][-sh]: logcheckd -a reset

logcheckd - a log file checking and reporting daemon, by Chr. Walther

initialize logcheckd daemon ...
  clear state information ..... done.
  clear collect data ..... done.
  clear saved reports ..... done.
done.

```

SEE ALSO

edrcintro(1), ecrontab(1), osid(3), crontab(1), lgcheckd(1m), lgcpattern(3), logcheckd.cfg(4), logcheckd.interface(3), logcheckd.pattern(4), logcheckd.style(4), tpl(1)

NOTES

-

BUGS

-

AUTHOR

logcheckd was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcheckd.cfg – configuration file for logcheckd and lgcheckd

SYNOPSIS

edrc/etc/logcheckd.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **logcheckd** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The fileformat for the '**GENERAL OPTIONS**' and the '**REPORT OPTIONS**' is **OPTION=VALUE**

Between the **OPTION**, the **=** and the **VALUE** are no spaces.

You should not comment out any **OPTION**. If you like to use default settings simply do not specify a **VALUE**.

For the '**LOGFILES OPTIONS**', the fileformat is *field ; field ; field ; field ; field ; field ;* (ensure, that between the *field* and the separator **;** are no spaces).

OPTIONS**GENERAL OPTIONS****USE_LEGACY_COMMAND**

If this setting is set to *True* the **logcheckd** command is started, if it is set to **False** the new variant **lgcheckd** is started when invoking the **logcheckd** command.

Example: USE_LEGACY_COMMAND=False

Default: USE_LEGACY_COMMAND=True

LOG

Log output dir of logcheckd. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: LOG=/var/opt/ACME/log

Default: LOG=var/log

LOCKDIR

Lock dir of logcheckd. If you specify a relative path name the path is relative to the root of the EDRC installation. In General it is not recommended to set the lockdir within EDRC, locate it in a system own directory.

Example: LOCKDIR=/var/run/logcheckd

Default: LOCKDIR=var/lock

CHECK_INTERVAL

Interval of **logcheckd** in seconds.

Example: CHECK_INTERVAL=30

Default: CHECK_INTERVAL=10

ANALYSIS_SUSPENSION

If the free space in the **COLLECTDIR** is not sufficient to analyze a logfile (the available free space must be 1.5 times the size of the logfile to be analyzed), the logfile analysis of the logfile is suspended for the number of seconds specified in this setting. When the **ANALYSIS_SUSPENSION** is elapsed, the free space is checked again. The logfile analysis will be suspended as long as the free space is not sufficient. When sufficient space is made available, the logfile analysis is resumed automatically.

Example: ANALYSIS_SUSPENSION=600

Default: ANALYSIS_SUSPENSION=300

MIN_FREESPACE

Minimum free disk space in percent (%) in the filesystem where the **COLLECTDIR** and the **REPORTDIR** resides. This is the watermark that is considered when checking the free space and which causes a analysis suspension or a "insufficient free space" message when generating a report. Beside to avoid file system fill ups, this setting is used to avoid monitoring alerts when a certain filesystem fill up level is reached.

Example: MIN_FREESPACE=5

Default: MIN_FREESPACE=15

INIT_MODE

Mode on initial startup when a logfile has not been analyzed before. Set *FromBeginning* to analyze the logfile from the beginning, set *FromEnd* to analyze logfile beginning at the file end.

Example: INIT_MODE=FromBeginning

Default: INIT_MODE=FromEnd

REPORTDIR

Directory where the LOGCHECK reports are written to. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: REPORTDIR=/dat/report/ACME/logcheck

Default: REPORTDIR=/var/logcheckd/report

CLEAN_REPORT

If this setting is set to *True* the report output file is removed after sending it.

Example: CLEAN_REPORT=True

Default: CLEAN_REPORT=False

COMPRESS_REPORTMAIL

If set to *True* the report is attached as a zipped file to the report mail.

Example: COMPRESS_REPORTMAIL=True

Default: COMPRESS_REPORTMAIL=False

COMPRESS_SUFFIX

Compressed attachment file suffix.

Example: COMPRESS_SUFFIX=.ACME.zip

Default: COMPRESS_SUFFIX=.zip

COMPANY

Company. This String will be added to the zipfile comment.

Example: COMPANY="ACME Information Business GmbH, Switzerland"

Default: COMPANY=""

REPORT_STYLE

Report style definition.

Example: REPORT_STYLE=WA2L.css

Default: REPORT_STYLE=WA2L.css

REPORT_NUMBER_START

First report number.

Example: REPORT_NUMBER_START=50000

Default: REPORT_NUMBER_START=10000

CLEAN_COLLECT

If this setting is set to *True* the report output file is removed after sending it.

Example: CLEAN_COLLECT=False

Default: CLEAN_COLLECT=True

COLLECTDIR

Directory where the log hits are collected to. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: COLLECTDIR=var/logcheckd/collect

Default: COLLECTDIR=var/logcheckd/collect

PATTERNDIR

Directory where the pattern files are saved. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: PATTERNDIR=var/logcheckd/pattern

Default: PATTERNDIR=var/logcheckd/pattern

CHECK_LEVELS

Check the listed levels only. This levels should to be defined in the pattern files to have an effect.

Example: CHECK_LEVELS=HIGH,MEDIUM,LOW,EXCLUDE

Default: CHECK_LEVELS=HIGH,MEDIUM,LOW

INTERFACECONFIGDIR

Directory where the interface configuration files are saved. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: INTERFACECONFIGDIR=var/logcheckd/iconfig

Default: INTERFACECONFIGDIR=var/logcheckd/iconfig

INTERFACES

Interfaces to be executed when a certain check level matches. The setting is a space separated list of entries in the format: *<interface>:<LEVEL1>,<LEVEL2>* .

Example: INTERFACES="mail:HIGH CAUnicenter:HIGH,MEDIUM"

Default: INTERFACES=""

ALERT_MAIL_TO

Recipient of the alert mail. Multiple recipients have to be specified as a comma separated list. If no **ALERT_MAIL_TO** is set, there is no alert mail sent on filesystem shortage and on resumption of log analysis.

Example: ALERT_MAIL_TO=fred.flintstone@acme.ch,support@acme.ch

Default: ALERT_MAIL_TO=""

ALERT_MAIL_SECTION

Section as displayed in the report mail subject line.

Example: ALERT_MAIL_SECTION="system alert"

Default: ALERT_MAIL_SECTION="system alert"

REPORT OPTIONS

REPORT_CUSTOMER

Customer.

Example: REPORT_CUSTOMER='server_environment -C'

Default: REPORT_CUSTOMER='server_environment -C'

REPORT_NAME

Name of the report.

Example: REPORT_NAME=LOGCHECK

Default: REPORT_NAME=LOGCHECK

REPORT_SECTION

Section as displayed in the report mail subject line.

Example: REPORT_SECTION=logcheck

Default: REPORT_SECTION=logcheck

REPORT_SERVER_ENVIRONMENT

Server environment where the report is generated.

Example: `REPORT_SERVER_ENVIRONMENT='server_environment'`

Default: `REPORT_SERVER_ENVIRONMENT='server_environment'`

REPORT_SERVER_ENVIRONMENT_DESCRIPTION

Server environment description where the report is generated.

Example: `REPORT_SERVER_ENVIRONMENT_DESCRIPTION='server_environment -d'`

Default: `REPORT_SERVER_ENVIRONMENT_DESCRIPTION='server_environment -d'`

REPORT_MAIL_FROM

Sender of the report mail.

Example: `REPORT_MAIL_FROM=support@acme.ch`

Default: `REPORT_MAIL_FROM=${USER}@'hostname'`

REPORT_MAIL_TO

Recipient of the report mail. If multiple recipients have to be specified as comma separated list.

Example: `REPORT_MAIL_TO=fred.flintstone@acme.ch,support@acme.ch`

Default: `REPORT_MAIL_TO=""`

REPORT_MAIL_CC

Carbon copy recipient of the report mail. If multiple recipients have to be specified as comma separated list.

Example: `REPORT_MAIL_CC=donald.duck@acme.ch,daisy.duck@acme.ch`

Default: `REPORT_MAIL_CC=""`

REPORT_MAIL_BCC

Blind carbon copy recipient of the report mail. If multiple recipients have to be specified as comma separated list.

Example: `REPORT_MAIL_BCC=dagobert.duck@acme.ch`

Default: `REPORT_MAIL_BCC=""`

REPORT_MAIL_CONFIRMATION_TO

Recipient of the confirmation mail.

Example: `REPORT_MAIL_CONFIRMATION_TO=support@acme.ch`

Default: REPORT_MAIL_CONFIRMATION_TO=""

LOGFILES OPTIONS

Logfiles. List of logfiles to be checked for patterns. Only existing files or files that once existed during a reporting period will be listed in the report. It is not needed to restart the logcheckd when logfile entries are modified, added or removed. Pattern file content can also be modified during logcheckd runtime.

The format of the logfiles specification is:

resolve_mode ; *patternfile* ; *res* ; *res* ; *res* ; *logfile* ;

(ensure, that between the *field* and the separator ; are no spaces)

where the fields have the following content:

resolve_mode

how to resolve the filename:

eval evaluate the filename dynamically. The variables \$HOSTNAME, \$TODAY, \$YEAR, \$MONTH and \$DAY are allowed to be used in filenames in the **eval** mode.

dflt take the filename precisely as specified

patternfile pattern file. The pattern files are located in the **var/logcheckd/pattern** directory. If an operating system dependent pattern file with the name **<patternfile>.<OSID>** exists (e.g. **Cron.Solaris**), this file will be used for logfile analysis, else the pattern file as specified (e.g. **Cron**) will be used.

res reserved for future use, leave it empty.

logfile logfile to analyze. The logfile has to be specified with an absolute path name.

Example:

```
dflt;WA2Ledrc;;;/opt/edrc/var/log/edrc.log;
dflt;All;;;/etc/rc.log;
eval;MCSG;;;/etc/cmcluster/*/*.cntl.log;
dflt;sudo;;;/var/adm/sudo.log;
dflt;Samba;;;/var/opt/samba/log.smbd;
dflt;syslog;;;/var/adm/syslog/syslog.log;
```

FILES

var/samples/templates/logcheck.cfg

Template configuration file for **logcheckd**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **osid(3)**, **lgcheckd(1m)**, **logcheckd(1m)**, **logcheckd.interface(3)**, **logcheckd.pattern(4)**, **logcheckd.style(4)**, **logtail(1)**, **server_environment(3)**, **tpl(1)**

NOTES

-

BUGS

-

AUTHOR

logcheckd.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcheckd.interface – interfaces for logcheckd and lgcheckd

SYNOPSIS

edrc/lib/logcheckd/interface/<interfacefile>

AVAILABILITY

WA2L/edrc

DESCRIPTION

the **INTERFACE** mechanism of **logcheckd** and **lgcheckd** allows **logcheckd** and **lgcheckd** to communicate to other programs for a matching log entry.

The **INTERFACE** receives the matching log entry thru **stdin**, the following environment variables are exported to the **INTERFACE**: **\$APPROOT**, **\$SERVER_ENVIRONMENT_NAME**, **\$SERVER_ENVIRONMENT_CUSTOMER**, **\$SERVER_ENVIRONMENT_DESCRIPTION**, **\$INTERFACE_CFG**, **\$TIMESTAMP**, **\$LOGFILE**, and **\$LEVEL**.

OPTIONS

-

ENVIRONMENT

\$INTERFACE_CFG

configuration file of the **INTERFACE**.

\$TIMESTAMP

timestamp of the matching logfile entry.

\$LOGFILE

name of the logfile containing the matching logfile entry.

\$LEVEL

matched level as defined in the **logcheckd.pattern**(4) file.

\$APPROOT

root directory of the WA2L/edrc installation as returned by the **approot**(3) command.

\$SERVER_ENVIRONMENT_NAME

server environment name as returned by the **server_environment**(3) command.

\$SERVER_ENVIRONMENT_CUSTOMER

server environment customer as returned by the **server_environment(3)** command.

\$SERVER_ENVIRONMENT_DESCRIPTION

server environment description as returned by the **server_environment(3)** command.

EXIT STATUS

0 always.

The exit status of an interface is not resolved by the **logcheckd** and **lgcheckd**.

FILES

edrc/var/logcheckd/iconfig/<interfacefile>.cfg

configuration file for the interface allowing to write a configurable interface.

EXAMPLES**1. Simple example interface:**

This example **INTERFACE** saves all matching log entries to an output file that is configured in the **edrc/var/logcheckd/iconfig/Example.cfg**.

Interface configuration (**edrc/var/logcheckd/iconfig/Example.cfg**):

```
#
# logcheckd/iconfig/Example.cfg - configuration for logcheckd interface
#
# [00] 23.05.2009 CWa Initial Version
#

# Output file for interface
#
INTERFACE_OUTPUTFILE=/tmp/$EDRC_SCRIPTNAME.out
```

Interface implementation (**edrc/lib/logcheckd/interface/Example**):

```
#!/bin/sh
#
# Example - logcheckd interface: simple Example
#
# [00] 23.05.2009 CWa Initial Version
# [01] 28.05.2009 CWa ++
#

# Const
. $INTERFACE_CFG
Outputfile=$INTERFACE_OUTPUTFILE
```

```

# run -- run the interface command
#
run() {
    cat <<EOM >> $Outputfile
--
    TIMESTAMP=$TIMESTAMP
    LEVEL=$LEVEL
    LOGFILE=$LOGFILE
    EOM

    cat - >> $Outputfile
} # run

```

```

# Main -- Main
#
main() {
    case $LEVEL in
        HIGH)
            run
            ;;
        *)
            run
            ;;
    esac
} # main
main

```

2. Other example interfaces:

See directory: **edrc/lib/logcheckd/interface/**.

SEE ALSO

edrcintro(1), **lgcheckd(1m)**, **logcheckd(1m)**, **logcheckd.cfg(4)**, **logcheckd.pattern(4)**

NOTES

-

BUGS

-

AUTHOR

logcheckd.interface was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcheckd.pattern – patternfile for logfile analysis in logcheckd and lgcheckd

SYNOPSIS

edrc/var/logcheckd/pattern/<patternfile>

AVAILABILITY

WA2L/edrc

DESCRIPTION

The pattern file is used to define the analysis of logfiles.

FILEFORMAT

The pattern file is a list of regular expression patterns divided into sections. Empty lines and lines starting with a # are considered as comments.

[SECTION_1]

pattern_11
pattern_12
pattern_1n

[SECTION_2]

pattern_21
pattern_22
pattern_2n

[SECTION_n]

pattern_n1
pattern_n2
pattern_nn

The sections and patterns are processed in the sequence listed. If a pattern matches for a logfile entry, the remaining patterns are not processed any more. Therefore it enhances performance if patterns that will match more often are listed at the beginning of a section, if possible.

Hint: a section with the identical name can appear multiple times in a pattern file.

The patterns are POSIX 1003.2 regular expressions as processed by **egrep**(1).

All logfile entries that are not matched are ignored and are counted as IGNORED.

The special section **[EXCLUDE]** can be used to exclude logfile entries from analysis. Excluded logfile

entries are not reported in the 'Logfiles' section of the LOGCHECK report, but are counted and reported in the 'Overview' section.

Because the sections are processed by **logcheckd(1m)** and **lgcheckd(1m)** in the order as defined in the pattern file, the **[EXCLUDE]** section will often be the first section in the pattern file to have the wanted effect.

The **[EXCLUDE]** section also enables you to reuse rule files already defined for the **logcheck(1)** command written by Craig H. Rowland, <crowland@psionic.com>.

The advantage of using the **[EXCLUDE]** section is, that the current logfile can be used as information baseline and all entries that are not significant to be reported can be excluded; all remaining entries can be classified, the pattern **.*** can be used to catch all remaining entries. Doing this unknown entries, which might be of interest, are not missed.

See also **EXAMPLES** section to see how to use the **[EXCLUDE]** section.

OPTIONS

-

EXAMPLES

See also **logcheckd(1m)** and **lgcheckd(1m)**.

1) example pattern file

See explanation of this pattern file in **logcheckd(1m)** and **lgcheckd(1m)** example 1) in section **EXAMPLES**.

```
#
# logcheckd/pattern/su - logfile analysis pattern file for: su
#
# [00] 25.04.2008 CWa Initial Version
#

[EXCLUDE]

.* + .* root-.*$

[HIGH]

.* - .*-root$

[MEDIUM]

.* + .*-root$
.* -

[LOW]
```


[VERIFY]

. *

SEE ALSO

edrcintro(1), **egrep(1)**, **lgcheckd(1m)**, **lgcheckd.cfg(4)**, **lgcpattern(3)**, **logcheckd(1m)**, **logcheckd.cfg(4)**, **logtail(1)**, **regexintro(4)**, **POSIX 1003.2** section 2.8 (Regular Expression Notation)

NOTES

use **lgcpattern(3)** to help defining a pattern file.

BUGS

-

AUTHOR

logcheckd.pattern was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcheckd.style – stylefile for logfile analysis in logcheckd

SYNOPSIS

edrc/lib/logcheckd/style/<stylefile>.css

AVAILABILITY

WA2L/edrc

DESCRIPTION

The style file is used to define the look of the LOGCHECK report.

It enables the user to define the own look of a report if needed.

The style file is included directly into the LOGCHECK report to ensure that the style definition takes effect independent of the file location.

With the **REPORT_STYLE** setting in the **logcheckd.cfg** config file it is possible to define a style sheet different from the default.

FILEFORMAT

The style file format complies to the definitions of the cascading style sheets (css).

OPTIONS

Below the style definitions special to the LOGCHECK report are explained.

REPORT

h1 Report title.

h2 Section titles:

h2.general Title of section '**General**'.

h2.overview Title of section '**Overview**'.

h2.logfiles Title of section '**Logfiles**'.

div.created
text in the footline of the report that contains the 'created by ...' text.

GENERAL SECTION
pre.general
Text in the '**General**' section.

OVERVIEW SECTION
table.overview
Overview table.

table.overview th.current
Table header of '**Current (hits)**' counter rows.

table.overview th.current_normalized
Table header of '**Current (hits/day)**' counter rows.

table.overview th.last
Table header of '**Last (hits/day)**' counter rows.

table.overview th.trend
Table header of '**Trend (hits/day)**' counter rows.

table.overview th.logfile
Table header of '**Logfile**' rows.

table.overview th.gap
Gap between overview table header columns.

table.overview td.current
Table cell of '**Current (hits)**' counter rows.

table.overview td.current a.marked
Current hit counts that are bigger then zero.

table.overview td.current a.normal
Current hit counts that are equal to zero.

table.overview td.current_normalized
Table cell of '**Current (hits/day)**' counter rows.

table.overview td.last
Table cell of '**Last (hits/day)**' counter rows.

table.overview td.trend
Table cell of '**Trend (hits/day)**' counter rows.

table.overview td.logfile

Table cell of '**Logfile NAME**' rows.

table.overview td.logsize

Table cell of '**Logfile SIZE**' rows.

table.overview td.gap

Gap between overview table data columns.

LOGFILES SECTION

h3 Logfile names in the '**Logfiles**' section.

h4 Level titles in the '**Logfiles**' section.

pre.logfiles

Selected logfile content in the.

pre.logfiles span.timing

Timing information in the selected logfile content.

pre.logfiles span.separator

Separator between timing information and the selected logfile content.

FILES**edrc/lib/logcheckd/style/WA2L.css**

Default style definition for the LOGCHECK report. Do not change this file if you like to change the report look, create an own **.css** file, otherwise your changes will be lost after an upgrade of WA2L/edrc.

EXAMPLES

-

SEE ALSO

edrcintro(1), **logcheckd(1m)**, **logcheckd.cfg(4)**, **CSS Kochbuch, 2005, O'Reilly, Schmitt Christopher, ISBN 3-89721-397-4**

NOTES

-

BUGS

-

AUTHOR

logcheckd.style was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcut – shorten logfiles

SYNOPSIS

edrc/bin/logcut [**-h** | **-p**]

logcut [**-s** *size*] **-c**

AVAILABILITY

WA2L/edrc

DESCRIPTION

logcut is used to shorten logfiles on a UNIX/Linux system. The logfile is shortened if a configurable row limit is reached. This limit can be set for each logfile.

To avoid to fill up a file system with logfiles this command may be placed in crontab and scheduled to run on a regular basis. See the **EXAMPLES** section for examples.

OPTIONS

-h the usage message.

-p print logfiles report.

-c cut/shorten the logfiles.

-s *size* size in rows. If specified, this command line option overrides the **LOGSIZE=** and **LOG[n]=** setting in the configuration file.

EXAMPLES

1. crontab entry to run logcut in a two hour interval:

```
# run logcut in a 2 hour interval
#
0 0,2,4,6,8,10,12,14,16,18,20,22 * * * [ -x /opt/edrc/bin/logcut ] \
&& /opt/edrc/bin/logcut -c >/dev/null 2>&1
```

FILES

etc/logcut.cfg
configuration file of **logcut** .

var/log/logcut.log
Logfile of **logcut** .

RETURN STATE

- | | |
|---|--|
| 0 | No error. |
| 2 | configuration file <i>logcut.cfg</i> does not exist. |
| 4 | usage listed. |
| 5 | program aborted (Ctrl-C, ...). |

SEE ALSO

edrcintro(1), **logcut.cfg(4)**

NOTES

logcut is derived from the **emlogcut** comand which is part of the EM-1/emtools module within EM-1, the EVENT-MANAGER-1 software package developed by Christian Walther. With EM-1 you can manage and resolve sports events. This is a database application based on Linux and PostgreSQL and provides an ASCII and a WEB interface, as well as many functions to manage a sports event and calculate rank lists and bills in an efficient way. For questions about EM-1 contact the author of EM-1: wa2l@users.sourceforge.net .

BUGS

-

AUTHOR

logcut was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logcut.cfg – configuration file for logcut

SYNOPSIS

edrc/etc/logcut.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **logcut** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You have to set all settings, no default values apply.

OPTIONS**LOGSIZE**

Default logfile length in rows if no specific length is specified in the **LOG[n]** settings.

Example: LOGSIZE=20000

Default: LOGSIZE=

LOG[n] Logfile(s) to shorten. The logfiles have to be specified as an absolute path name. Each logfile to be shorten needs a **LOG[n]** entry with an own number (*n*).

The format of the **LOG[n]** entries is: **LOG[n]=size:logfile**

Example: LOG[01]=100000:~edrc/var/log/edrc.log

Default: LOG[01]=

SEE ALSO

logcut(1), **edrcintro(1)**

NOTES

-

BUGS

No default values apply, therefore ensure to specify all settings.

AUTHOR

logcut.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

loggrep – grep in remote- or local (time-stamped) log file(s)

SYNOPSIS

edrc/bin/loggrep [**-h** | **-p**]

loggrep *regex file ...*

loggrep *regex* [[*user@*]*host*:]*file ...*

loggrep [*options*] [**-f** "*from_ts*"] [**-t** "*to_ts*"] [*egrep_options*] *regex* [[*user@*]*host*:]*file ...*

loggrep [*options*] [**-f** "*from_ts*"] [**-t** "*to_ts*" **-t** "*to_ts*"] [*egrep_options*] *regex* [[*user@*]*host*:]*file ...*

lgrep ...

options ::= [**-l** *localuser*] [**-M**] [**-s** *num*] [**-NUM**] [**-T** *id*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

grep selected time range (from/to) in log *files*.

If the log file has time stamps, the data between a *from* and *to* time stamp is selected and **grep**ed by the specified *regex*.

When there are no time stamps in the log file, or the time stamp format is not recognized, the whole log file data is **grep**ed

The first **97** lines of the log file are used to resolve the time stamp format used in a particular file.

Beside text log file(s), **loggrep** can handle file(s) which are **zipped** (**.zip**), **compressed** (**.Z**), **gzipped** (**.gz**), **bzip2ed** (**.bz**, **.bz2**, **.bzip2**) or **xzed** (**.xz**) directly.

To increase performance on scanning large log files which are bigger then **100 MByte**, only the last **1'000'000** lines are analysed. To force to nevertheless analyse the whole file, specify **-0** in the **-NUM** option.

FROM/TO DATA SELECTION

The **-f** "*from_ts*" and **-t** "*to_ts*" time stamps have to be specified in military format ("**2021-01-18 20:57:10**").

The following abbreviations and defaults apply (examples when executed on **Monday, 2021-01-18**):

| specified from/to_ts | resulting from_ts | resulting to_ts |
|----------------------|---------------------|---------------------|
| | 2021-01-18 00:00:00 | 2021-01-18 23:59:59 |
| 13:45 | 2021-01-18 13:45:00 | 2021-01-18 13:45:59 |
| 13:45:55 | 2021-01-18 13:45:55 | 2021-01-18 13:45:55 |
| 16 | 2021-01-16 00:00:00 | 2021-01-16 23:59:59 |
| 01-16 | 2021-01-16 00:00:00 | 2021-01-16 23:59:59 |
| 05-30 | 2020-05-30 00:00:00 | 2020-05-30 23:59:59 |
| 2020 | 2020-01-01 00:00:00 | 2020-12-31 23:59:59 |
| 2020-03 | 2020-03-01 00:00:00 | 2020-03-31 23:59:59 |
| 2020-03-27 | 2020-03-27 00:00:00 | 2020-03-27 23:59:59 |
| 2020-03-27 13 | 2020-03-27 13:00:00 | 2020-03-27 13:59:59 |
| 2020-03-27 13:45 | 2020-03-27 13:45:00 | 2020-03-27 13:45:59 |
| 2020-03-27 13:45:12 | 2020-03-27 13:45:12 | 2021-03-27 13:45:12 |
| yesterday | 2021-01-17 00:00:00 | 2021-01-17 23:59:59 |
| mon | 2021-01-11 00:00:00 | 2021-01-11 23:59:59 |
| tue | 2021-01-12 00:00:00 | 2021-01-12 23:59:59 |
| wed | 2021-01-13 00:00:00 | 2021-01-13 23:59:59 |
| thu | 2021-01-14 00:00:00 | 2021-01-14 23:59:59 |
| fri | 2021-01-15 00:00:00 | 2021-01-15 23:59:59 |
| sat | 2021-01-16 00:00:00 | 2021-01-16 23:59:59 |
| sun | 2021-01-17 00:00:00 | 2021-01-17 23:59:59 |

LOG FILE TIME STAMP FORMATS

The following time stamps are recognized (per log file):

- 1) 2021-01-30 23:59:59
- 2) 2021/01/30-23:59:59
- 3) Jan 30 23:59:59
- 4) ---- 2021-01-30 23:59:59 ----
- 5) 12/16-11:34:21
- 6) Nov 18, 2019 9:44:43 PM
Apr 15, 2020 10:50:51 AM
- 7) End-Date: 2021-01-15 10:06:29
Start-Date: 2021-01-15 10:06:35
- 8) [18/Jan/2021:10:51:23 +0100]
- 9) 2021/01/30 23:59:59
- 10) ----BEGIN: Mon Nov 18 19:51:51 UTC 2019
----END: Mon Nov 18 19:51:52 UTC 2019
- 11) 2020-10-27T17:44:21

- 12)** #1612518675
- 13)** 2021-01-30-23:59:59
- 14)** [Tue Aug 27 10:56:41 2019]
[Tue Aug 27 10:56:41.1234 2019]
- 15)** Thu Feb 4 12:27:12 2021
- 16)** -80238 | Fri Feb 5 12:27:11 2021
447752 | Fri Feb 5 12:27:12 2021
- 17)** update-alternatives 2020-10-27 22:50:52
- 18)** 0156 01/31/21 12:23:56
0157 01/31/21 12:24:07
- 19)** ---- Rules Applied on Monday 2021-01-11 12:25:48
- 20)** [UTC 01/31/21 11:22:32]
- 21)** Log started: 2021-01-15 10:06:29
Log ended: 2021-01-15 10:06:35
- 22)** Log time: 2018-08-18 21:10:09
- 23)** error 2021-02-13 06:12:51
info 2021-02-13 06:12:51
- 24)** ERROR: apport (pid 593) Fri Feb 1 23:09:19 2019

OPTIONS

- h** usage message.
- p** print the timestamp formats known by **loggrep**, some example files where that timestamp format has been discovered and the regular expression that is used to identify the timestamp.
- l *localuser*** use the local user *localuser* to initiate the connection.
- s *num*** use the first *num* lines of the logfile for timestamp format detection.
- NUM** analyse the last *NUM* lines, instead of the whole file.

On large log files that exceed **100 MByte**, only the last **1'000'000** rows are scanned, whereas the first **4007** lines are here used to resolve the time stamp format.

To force to scan nevertheless the whole log file, specify `-0` here.

-f "*from_ts*"

start of data selection in military (=ISO) format, an abbreviated date or time specification as listed in the table in section **DESCRIPTION** or a past weekday (as: **yesterday**, **mon**, **tue**, **wed**, **thu**, **fri**, **sat**, **sun**).

Default is the current date at **00:00:00** (example: "**2021-01-18 00:00:00**").

-t "*to_ts*"

end of data selection in military (=ISO) format, an abbreviated date or time specification as listed in the table in section **DESCRIPTION** or a past weekday (as: **yesterday**, **mon**, **tue**, **wed**, **thu**, **fri**, **sat**, **sun**).

Default is current date at **23:59:59** (example: "**2021-01-18 23:59:59**").

-M

print meta data of **loggrep** results to file descriptor **3**.

Example:

```
LOGGREP_FROM="2020-03-27 13:45:00"
LOGGREP_TO="2020-03-27 18:00:00"
LOGGREP_COUNT="68312"
LOGGREP_BEGIN="2020-03-27 13:50:18"
LOGGREP_END="2020-03-27 17:58:36"
LOGGREP_DURATION="000:04:13:24"
```

This output can be used to set variables in a script:

```
#!/bin/sh

loggrep -M -f "2020-03-27 13:45:00" -t "2020-03-27 18:00:00" \
    starting ~edrc/var/log/edrc.log 3>/tmp/meta.out

. /tmp/meta.out
echo "ENTRIES: $LOGGREP_COUNT between $LOGGREP_BEGIN and $LOGGREP_END"
```

egrep_options

options of the **egrep**(1) command.

If options of the **egrep**(1) have a second argument, specify the argument without a space (eg: specify **-m5** and not **-m 5**).

See **egrep**(1) for a description of the available **egrep** options.

regex

regular expression *PATTERN* as understood by **egrep** respectively **grep -E**.

[[user@]host:]file

remote- or local file.

If a `-` is specified as a *file* option, data is read from **stdin**.

Beside text files, the specified *file* can also be **zipped** (**.zip**), **compressed** (**.Z**), **gzipped** (**.gz**), **bzip2ed** (**.bz**, **.bz2**, **.bzip2**) or **xzed** (**.xz**).

ENVIRONMENT

\$LOGGREG_FROM

from_ts as specified on the command line. However, the command line option **-f** "*from_ts*" has preference.

\$LOGGREG_TO

to_ts as specified on the command line. However, the command line option **-t** "*to_ts*" has preference.

EXIT STATUS

- 0** *regex* is found in at least one of the listed files. no error.
- 1** *regex* is not found in any of the listed files.
- 2** error occurred.
- 4** usage printed.
- 5** command has been aborted.
- 11** could not claim a temporary directory in **/tmp/**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **cat**(1), **edrcsetup**(1m), **egrep**(1), **grep**(1), **logcat**(1), **rcat**(1), **regexintro**(4), **ssh-exec**(1), **ssh-exec.cfg**(4)

NOTES

loggrep uses **rcat**(1) internally to get the remote- and local files.

A call to **lgrep** ... is identical to a call to the **loggrep** ... command.

BUGS

-

AUTHOR

loggrep was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logrotate – rotate logfiles

SYNOPSIS

edrc/bin/logrotate [**-h**]

logrotate -f *file* [**-r** *number_of_saves*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

logrotate rotates logfiles on the system. The permissions and the file handle of the logfile (*file*) are preserved.

The *file* is saved to *file.<number>* where the **<number>** counts from **1** to *number_of_saves*.

The newest version of the file is the one having the lowest number, the oldest file has the highest number.

If the specified *file* does not exist, the already saved versions are not rotated.

OPTIONS

-h usage message.

-f *log_file* logfile to be rotated.

-r *number_of_saves*
number of rotated files that are kept on the filesystem.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 specified *file* to be rotated does not exist.

- 2** cannot write to logfile of **logrotate**.
- 3** configfile **logrotate.cfg** does not exist.
- 4** usage printed.

FILES

etc/logrotate.cfg

configuration file of **logrotate**, see **logrotate.cfg(4)** for more information.

var/log/logrotate.log

default logfile of **logrotate**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **logrotate.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

logrotate was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logrotate.cfg – configuration file for logrotate

SYNOPSIS

edrc/etc/logrotate.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **logrotate** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**LOGBASE**

basedir of the logrotate logfile (**logrotate.log**).

Example: LOGBASE='homedir ACME'/log

Default: LOGBASE='approot'/var/log

DEFAULT_SAVES

Default number of file generations kept. This setting applies if the **-r number_of_saves** option is not specified on the command line.

Example: DEFAULT_SAVES=5

Default: DEFAULT_SAVES=10

SEE ALSO

logrotate(1), **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

logrotate.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logtail – tail a log file out of a list of defined logfiles

SYNOPSIS

edrc/bin/logtail [**-h**]

logtail [*select*]

logtail [**-t**] [**-c** | **-C**] [*select*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

give a selection of common logfiles to be tailed (**tail -f *selected_logfile***). The files to be listed in the selection list can be defined in a user dependent or system wide configuration file.

The config file can list logfiles using the wild cards as known from the **ls(1)** command. The list is resolved prior to each start of **logtail** and only the files that exist on the system and are readable by the user are listed in the selection list.

This enables the user to have an identical configuration file for all servers and operating systems.

To load a log file into a viewer (**view**), instead of the continuous file output of **logtail**, use the **logview(1)** command.

OPTIONS

- t** add a high resolution timestamp to the output allowing timing information display on logfiles without timestamps.
- c** colorize **tailed** output based on **logcheckd.pattern(4)** files.
- C** prepend [*LEVEL*] to the colorizes **tailed** output based on **logcheckd.pattern(4)** files.
- select* select files from logfile list.

ENVIRONMENT**\$APPROOT**

root directory of the application as resolved by **approot(3)**. This variable can be used in the log files configuration files to point to the root of the WA2L/edrc installation. Therefore for example the **~edrc/var/log/edrc.log** definition can be replaced with

\$APPROOT/var/log/edrc.log with the result that **logview** also lists the logfile when no **edrc** user is defined on the system.

\$LOGTAIL_LEVELMODE

set the appearance of the [*LEVEL*] effective when using the **-C option**.

Possible settings are: **light**, **bold**, **invers**, **red**, **green**, **magenta**, **blue**, **cyan** and **yellow**, whereas the default is **bold**.

EXIT STATUS

0 no error.

FILES

\$HOME/.log_files

definition of files to be listed in the selection. Only the root user should list the files in this config file.

\$HOME/.mylog_files

definition of files to be listed in the selection. For non-root users, this file should be used to define the log files instead of **.log_files** due to the preservation of *.my** files when using the **mkuser(3)** and/or **makeuser(3)** commands.

etc/log_files.cfg

system wide definition of log files. This file is read when the files **\$HOME/.log_files** and **\$HOME/.mylog_files** do not exist.

etc/lgcheckd.cfg

configuration file of **lgcheckd(1m)** to resolve the pattern file for a selected log file.

etc/logcheckd.cfg

configuration file of **logcheckd(1m)** to resolve the pattern file for a selected log file. This file is read, if the **etc/lgcheckd.cfg** does not exist.

var/logcheckd/pattern/

pattern files.

var/logcheckd/pattern/logtail

default pattern file, if no pattern file could be resolved.

EXAMPLES

1) common usage

Configuration file example:

```
#
# log_files.cfg - configuration file for logtail and logview
#
# [00] 08.08.2004 CWa Initial Version
#
$APPROOT/var/log/edrc.log
$APPROOT/var/log/pack.log
$APPROOT/var/log/patchinstall.log
$APPROOT/var/log/filedist.log
$APPROOT/var/log/rcmd.log
$APPROOT/var/log/sysconfig.log
$APPROOT/var/log/shttpd_*.log
/etc/cmcluster/*/*.log
/etc/shutdownlog
~ACME/log/sap_instance.log
/opt/patrol/log/*.log
/var/adm/cron/log
/var/log/boot.log
/var/log/cron
/var/log/cups/access_log
/var/log/cups/error_log
/var/log/dmesg
/var/log/httpd/*
/var/log/lastlog
/var/log/maillog
/var/log/messages
/var/adm/messages
/var/log/samba/log.nmbd
```

Example usage:

```
[ /home/barney ]
[ barney@acme001 ][bash]: logtail

1) /opt/edrc/var/log/edrc.log
2) /opt/edrc/var/log/pack.log
3) /opt/edrc/var/log/patchinstall.log
4) /opt/edrc/var/log/sysconfig.log
5) /opt/ACME/log/sap_instance.log
6) /var/log/boot.log
7) /var/log/cron
8) /var/log/dmesg
9) /var/log/httpd/access_log
10) /var/log/httpd/access_log.1
11) /var/log/httpd/access_log.2
12) /var/log/httpd/access_log.3
13) /var/log/httpd/access_log.4
14) /var/log/httpd/error_log
15) /var/log/httpd/error_log.1
:
41) /var/opt/ACME/log/useradm.log
42) /var/opt/ACME/log/watchdog.log
43) /var/opt/samba/log.smb.rh7mzv7t001
select log:6
```

```
logtail-INFO: tail -f /var/log/cron
Jan 14 14:01:00 rh7mzv7t001 CROND[27550]: (root) CMD (run-parts /etc/cr
Jan 14 15:01:00 rh7mzv7t001 CROND[27913]: (root) CMD (run-parts /etc/cr
Jan 14 16:01:00 rh7mzv7t001 CROND[28276]: (root) CMD (run-parts /etc/cr
Jan 14 18:01:00 rh7mzv7t001 CROND[29478]: (root) CMD (run-parts /etc/cr
```

SEE ALSO

edrcintro(1), **logcheckd.pattern(4)**, **lgcheckd.cfg(4)**, **logcheckd.cfg(4)**, **logview(1)**, **log_files.cfg(4)**, **tail(1)**, **tpl(1)**

NOTES

-

BUGS

-

AUTHOR

logtail was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logview – view a log file out of a list of defined logfiles

SYNOPSIS

edrc/bin/logview [**-h**]

logview [*select*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

give a selection of common logfiles to be viewed (**view** *selected_logfile*). The files to be listed in the selection list can be defined in a user dependent or system wide configuration file.

The config file can list logfiles using the wild cards as known from the **ls**(1) command. The list is resolved prior to each start of **logview** and only the files that exist on the system and are readable by the user are listed in the selection list.

This enables the user to have an identical configuration file for all servers and operating systems.

To continuously print a log file (**tail -f**), instead of loading the file into the viewer of **logview**, use the **log-tail**(1) command.

OPTIONS

-h print usage message,

select select files from list of files.

ENVIRONMENT**\$APPROOT**

root directory of the application as resolved by **approot**(3). This variable can be used in the log files configuration files to point to the root of the WA2L/edrc installation. Therefore for example the **~edrc/var/log/edrc.log** definition can be replaced with **\$APPROOT/var/log/edrc.log** with the result that **logview** also lists the logfile when no **edrc** user is defined on the system.

EXIT STATUS

0 no error.

4 usage message printed.

FILES

\$HOME/.log_files

definition of files to be listed in the selection. Only the root user should list the files in this config file.

\$HOME/.mylog_files

definition of files to be listed in the selection. For non-root users, this file should be used to define the log files instead of **.log_files** due to the preservation of *.my** files when using the **mkuser(3)** and/or **makeuser(3)** commands.

etc/log_files.cfg

system wide definition of log files. This file is read when the files **\$HOME/.log_files** and **\$HOME/.mylog_files** do not exist.

etc/exrc settings for the **view** read only editor.

EXAMPLES

1) common usage

Configuration file example:

```
#
# log_files.cfg - configuration file for logview and logview
#
# [00] 08.08.2004 CWa Initial Version
#
$APPROOT/var/log/edrc.log
$APPROOT/var/log/pack.log
$APPROOT/var/log/patchinstall.log
$APPROOT/var/log/filedist.log
$APPROOT/var/log/rcmd.log
$APPROOT/var/log/sysconfig.log
$APPROOT/var/log/shutdown.log
/etc/cmcluster/*/*.log
/etc/shutdownlog
~ACME/log/sap_instance.log
/opt/patrol/log/*.log
/var/adm/cron/log
/var/log/boot.log
/var/log/cron
/var/log/cups/access_log
/var/log/cups/error_log
/var/log/dmesg
/var/log/httpd/*
/var/log/lastlog
/var/log/maillog
/var/log/messages
```

```
/var/adm/messages
/var/log/samba/log.nmbd
```

Example usage:

```
[ /home/barney ]
[ barney@acme001 ][bash]: logview

1) /opt/edrc/var/log/edrc.log
2) /opt/edrc/var/log/pack.log
3) /opt/edrc/var/log/patchinstall.log
4) /opt/edrc/var/log/sysconfig.log
5) /opt/ACME/log/sap_instance.log
6) /var/log/boot.log
7) /var/log/cron
8) /var/log/dmesg
9) /var/log/httpd/access_log
10) /var/log/httpd/access_log.1
11) /var/log/httpd/access_log.2
12) /var/log/httpd/access_log.3
13) /var/log/httpd/access_log.4
14) /var/log/httpd/error_log
15) /var/log/httpd/error_log.1
:
41) /var/opt/ACME/log/useradm.log
42) /var/opt/ACME/log/watchdog.log
43) /var/opt/samba/log.smb.rh7mzv7t001
select log:6
```

```
1 Jan 14 14:01:00 rh7mzv7t001 CROND[27550]: (root) CMD (run-parts /et
2 Jan 14 15:01:00 rh7mzv7t001 CROND[27913]: (root) CMD (run-parts /et
3 Jan 14 16:01:00 rh7mzv7t001 CROND[28276]: (root) CMD (run-parts /et
4 Jan 14 18:01:00 rh7mzv7t001 CROND[29478]: (root) CMD (run-parts /et
~
~
```

SEE ALSO

edrcintro(1), logtail(1), log_files.cfg(4), tail(1), tpl(1)

NOTES

-

BUGS

-

AUTHOR

logview was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

log_files.cfg – log files definition for logfiles and logtail

SYNOPSIS

edrc/etc/log_files.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the log file definition for the **logview** and **logtail** commands.

FILEFORMAT

The fileformat is a plain list of log files, where wildcards, the tilde (~) and the variables documented in the **OPTIONS** section are supported.

The log files must not exist on the system, the **logview** and **logtail** commands resolve the existing log files and only presents the existing ones to the user; therefore the **log_files.cfg** can be kept identical on all systems.

Rows starting with a # are considered as comments. Empty lines are allowed, too.

OPTIONS**\$APPROOT**

root of the WA2L/edrc package installation, as returned by the **aproot(3)** command.

EXAMPLES

```
#
# log_files.cfg - configuration file for logtail and logview
#
# [00] 08.08.2004 CWa   Initial Version
#
$APPROOT/var/log/edrc.log
$APPROOT/var/log/pack.log
$APPROOT/var/log/patchinstall.log
$APPROOT/var/log/filedist.log
$APPROOT/var/log/rcmd.log
$APPROOT/var/log/sysconfig.log
$APPROOT/var/log/shttpd_*.log
/etc/cmcluster/*/*.log
/etc/shutdownlog
~ACME/log/sap_instance.log
```

```
/opt/patrol/log/*.log
/var/adm/cron/log
/var/log/boot.log
/var/log/cron
/var/log/cups/access_log
/var/log/cups/error_log
/var/log/dmesg
/var/log/httpd/*
/var/log/lastlog
/var/log/maillog
/var/log/messages
/var/adm/messages
/var/log/samba/log.nmbd
```

SEE ALSO

edrcintro(1), **logtail(1)**, **logview(1)**

NOTES

-

BUGS

-

AUTHOR

log_files.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lots – long term data save handling

SYNOPSIS

edrc/bin/lots [**-h**]

lots [**-c** *configfile*] **-a** (**collect** [**-d** *datalist*] | **save** | **execute**) [**-i** *identitylist*]

lots [**-c** *configfile*] **-a** (**lock** | **purge**)

lots [**-c** *configfile*] **-a** **clear** **-j** *jobname*

lots [**-c** *configfile*] **-a** (**list_action** | **list_session** | **list_collect** | **list_save** | **list_lock** | **list_purge** | **list_clear**) [**-f** *from_date*][**-t** *to_date*]

lots [**-c** *configfile*] **-a** **list_jobs** [**-d** *datalist*]

lots [**-c** *configfile*] **-a** (**list_datalist** | **list_schedule** | **list_volume**)

lots [**-c** *configfile*] **-a** (**print_job** **-j** *jobname* | **print_log** [**-f** *from_date*][**-t** *to_date*] | **print_logtail** | **print_session** **-s** *sessionname*)

AVAILABILITY

WA2L/edrc

DESCRIPTION

lots is used to copy data in an organized and automated fashion to a long term storage device. On this device the data is locked.

lots locks the data by setting the access time of the saved data to the future point in time (current day + *RETENTION*) and removes the write flag of all files and directories to be locked. If the long term data storage is a file system on a NetAPP filer having a 'SnapLock Enterprise' volume or a 'SnapLock Compliance' volume, the data cannot be removed from Unix until the *RETENTION* is expired. If **lots** is used on a normal file system not having this functionality, the root user can delete the data, but it is still ensured, that non-root users cannot view, list or even remove the saved data.

The source path where the data comes from, the permissions of the path and other properties are saved and recorded to ensure a data restore even if the user base has changed since the data save.

After the expiration of the *RETENTION* + *DATA_PURGE_LAG*, the **lots** command purges (=removes) the expired data and the disk space is freed up.

The following four steps are performed during the life cycle of a long term data save:

1) collect

This step is executed once for each *DATALIST* whose schedule in **schedule.dat** matches for the current day and whose *HOSTNAME* setting matches to the host where the **lots** command is

started.

A new job is created and the file and directory names to be saved as defined in the **datalist.dat** file are collected and stored in the newly created job. Be aware, that the **lots** release that created a job is always saved in the job, too (**VERSION** property when printing the job).

This job will in the following steps be passes thru several phases during the whole information life cycle. The job information can be displayed using the ' **lots -a print_job -j jobname** ' command.

In this step no data is saved. When the **collect** action is completed, the job is in the **save** phase.

If multiple schedule definitions of a **DATALIST** match at a certain day, only the schedule definition having the highest **RETENTION** value will be executed, this to use the disk space economically. A scheduled **DATALIST** is only executed once a day unless the **RETENTION** in the **schedule.dat** is increased.

A job that is in the **save** phase can be removed using the ' **lots -a clear -j jobname** ' command.

2) save

This step is executed for all jobs that are in the **save** phase based on the resolved hostname in the **collect** step.

In this step the files found in the **collect** step are copied to the long term data save location as defined in the **volume.dat** file. The save can be performed with or without compression. The compression method has to be defined in the **schedule.dat** file.

When the **save** action is completed, the job is in the **lock** phase.

A job that is in the **lock** phase can be removed using the ' **lots -a clear -j jobname** ' command. All data that has been saved will also be removed from the long term storage.

3) lock

This step is executed for all jobs that are in the **lock** phase, the **save** phase completed with a **SAVESTATE** of **OK** and the number of days as defined in the **DATA_LOCK_LAG** setting in the **lots.cfg** file are expired.

The lock delay enables the system administrative personnel to react on unwanted conditions, as: inconsistency of saved data, errors in definitions which cause too much data to be saved (=waste of disk space) and other reasons.

The **DATA_LOCK_LAG** should be defined carefully and the implications have to be known by the administrative personnel operating the long term data save setup. If, for instance, the **DATA_LOCK_LAG** is set to 7, the administrator has after the **save** seven days time to react on a malfunction and to correct it. But during this time it is also possible to delete the saved data, what can be a risk, too.

When a job is in the **lock** phase, but the previous phase (**save**) did not complete with a **SAVESTATE** equal to **OK**, the **lock** action for the related job will be skipped and the data will not be locked. This means, that the job will remain in the **lock** phase until it is cleared using the ' **lots -a clear -j jobname** ' command. This avoids the locking of saved data of unsuccessful **save** actions.

When the **lock** action is completed, the job is in the **purge** phase.

A job that is in the **purge** phase cannot be removed any more using the '**lots -a clear -j job-name**' command, because on the long term storage the data is locked and **lots** must wait for the expiration of the lock to remove the data related to the locked job.

4) purge

This step is executed for all jobs that are in the **purge** phase after *RETENTION* + *DATA_PURGE_LAG* is expired.

The **purge** action removes the data related to the job on the long term storage and the disk space is therefore freed up.

When the **purge** action is completed, the job is in the **expired** phase and will remain there for ever for reference and compliance proof purposes.

The **DATA_PURGE_LAG** has only to be defined greater than 0, if it is experienced, that the system clocks of the server(s) accessing the long term storage (using **lots**) and the 'Compliance Clock' of the long term data storage device are not completely in sync. The purge of the data is delayed by the number of days defined in this setting. This to ensure smooth data purging.

This four steps (**collect**, **save**, **lock**, **purge**) can be executed separately using the commands:

- 1) **lots -a collect**
- 2) **lots -a save**
- 3) **lots -a lock**
- 4) **lots -a purge**

but the normal case is using the '**lots -a execute**' command which executes all steps in one session in sequence.

In a productive automated setup, the '**lots -a execute**' command will normally be called via a **cron** entry that starts **lots** once each day.

It is supported to access the same long term storage in parallel from multiple hosts. In this case the **VARDIR** as defined in the **lots.cfg** file and the **lots.cfg** file itself should be located also centrally on the long term storage to allow the most convenient operations. It is anyway recommendable to put the **VARDIR** and the **lots.cfg** config file also on the long term storage device to have the configuration and state information separated from the server to always secure this information and to be independent from server crashes and server re-installations which might occur during the life cycle of the data stored for a long time on the long term data storage. For central secure setup as suggested here, see the **EXAMPLES** section.

Each invocation of '**lots -a action**' creates a session. A session represents all command output and can be displayed for verification and compliance proof. Session logs are kept forever and can be displayed using the '**lots -a print_session -s sessionname**' command. To evaluate the sessions related to a job, use the '**lots -a list_(action|collect|save|lock|purge|clear)**' command.

To resolve the data save location for a job of a certain *DATALIST*, use the commands:

- 1) **lots -a list_jobs [-d datalist]**
- 2) **lots -a print_job -j jobname**

The **lots** command currently does not provide a data restore interface. Therefore the restore of the saved files is performed using the normal operating system commands (**cp**, **cpio**, **unzip**, **bzip2 -d**, **gzip -d**,

uncompress) and after the restore the permissions have to be adjusted based on the information queried using the **print_job** action as described above also using the operating system commands (**chmod**, **chown**, **chgrp**).

OPTIONS

-h usage message. Here the revision of **lots** is also displayed.

-c *config_file*
configuration file.

-i *identitylist*
comma separated list of identities (=hostnames) of the **lots** command.
If this option is not specified and the environment variable **\$LOTS_IDENTITIES** is not set, the identities of the **lots** command are all hostnames under which the host where **lots** is started is known. Using the **-i** option it can be defined on which *HOSTNAME* settings in the **schedule.dat** file the **lots** command reacts.

Example, when **-i adm_oral_tst** is used:

```
:
resolve identities of this host ...
    adm_oral_tst
done.
:
```

Example, when the **-i** option is not used and the **\$LOTS_IDENTITIES** environment variable is not set (the *adm_oral_tst* hostname is a cluster package that is currently running on the host):

```
:
resolve identities of this host ...
    adm_oral_tst acme001 loghost localhost
done.
:
```

-a action:

collect [**-d** *datalist*] [**-i** *identitylist*]
collect data of scheduled *datalist*(s).

save [**-i** *identitylist*]
save data that has been collected to the long term storage.

lock
lock data on the long term storage from modification.

purge
remove data whose locks (retention) have been expired from long term storage.

clear -j *jobname*

remove a job and the data saved (if any). It is only possible to clear a job in the **save** and **lock** phases. When a job is cleared that has the biggest retention for a datalist for the current date, it is possible to perform another **collect** action of the related schedule.

execute [-i *identitylist*]

perform all four long term data save steps (**collect**, **save**, **lock**, **purge**) in one step.

list_action [-f *from_date*] [-t *to_date*]

list all performed actions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the actions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the actions are listed from the *from_date* to the end. If neither of the two dates are specified all actions are listed.

list_session [-f *from_date*] [-t *to_date*]

list all saved sessions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the sessions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the sessions are listed from the *from_date* to the end. If neither of the two dates are specified all sessions are listed.

list_collect [-f *from_date*] [-t *to_date*]

list only the **collect** actions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the **collect** actions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the **collect** actions are listed from the *from_date* to the end. If neither of the two dates are specified all **collect** actions are listed.

list_save [-f *from_date*] [-t *to_date*]

list only the **save** actions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the **save** actions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the **save** actions are listed from the *from_date* to the end. If neither of the two dates are specified all **save** actions are listed.

list_lock [-f *from_date*] [-t *to_date*]

list only the **lock** actions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the **lock** actions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the **lock** actions are listed from the *from_date* to the end. If neither of the two dates are specified all **lock** actions are listed.

list_purge [-f *from_date*] [-t *to_date*]

list only the **purge** actions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the **purge** actions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the **purge** actions are listed from the *from_date* to the end. If neither of the two dates are specified all **purge** actions are listed.

list_clear [-f *from_date*] [-t *to_date*]

list only the **clear** actions between the *from_date* and the *to_date*.

If the *from_date* is not specified, the **clear** actions are listed from the beginning to the *to_date*. If the *to_date* is not entered, the **clear** actions are listed from the *from_date*

to the end. If neither of the two dates are specified all **clear** actions are listed.

list_jobs [-d *datalist*]

list the jobs and the phase where the job is in. If the **-d *datalist*** is not specified, all jobs are listed.

list_datalist

list all valid datalist definitions. See also **datalist.dat(4)** for more information about the datalist format.

list_schedule

list all valid schedule definitions. See also **schedule.dat(4)** for more information about the schedule format.

list_volume

list all valid volume definitions. See also **volume.dat(4)** for more information about the volume format.

print_job -j *jobname*

print all properties of a certain job. Use this option to print the path of the saved data.

GENERAL JOB PROPERTIES:

PHASE phase the job is currently in.

HOSTNAME

hostname where the data of a matching schedule has been collected and saved.

VERSION

version of **lots** that created the related job.

JOBNAME

name of the job. This name has to be specified in the **-j** option when required.

DATALIST

scheduled datalist name of the job.

DATALIST_DESCRIPTION

free text description of the **DATALIST** as defined in the **datalist.dat** file at the time of job creation.

TIMEZONE

time zone as returned by **timezone(3)** at the time of job creation.

SCHEDULE

schedule in **schedule.dat** which matched at the time of job creation.

SCHEDULE_DESCRIPTION

free text description of the **SCHEDULE** as defined in the **schedule.dat** file at the time of job creation.

RETENTION

effective data retention in days as defined in the **schedule.dat** file at the time of job creation.

COMPRESSION

compression method of the saved data as defined in the **schedule.dat** file at the time of job creation.

COLLECTTIMESTAMP

timestamp (seconds since the epoch) when the data to be saved has been collected. This property will be seen when the job is in the phase **save**, **lock**, **purge** or **expired**.

COLLECTTIMEDAT

this is the human readable format of **COLLECTTIMESTAMP**.

COLLECTSTATE

state of the **collect** action. This property will be seen when the job is in the phase **save**, **lock**, **purge** or **expired**.

LOCKTIMESTAMP

timestamp (seconds since the epoch) when the data to be saved will be locked. This property will be seen when the job is in the phase **save**, **lock**, **purge** or **expired**.

LOCKTIMEDAT

this is the human readable format of **LOCKTIMESTAMP**.

LOCKSTATE

state of the **lock** action. This property will be seen when the job is in the phase **purge** or **expired**.

SAVETIMESTAMP

timestamp (seconds since the epoch) when the data has been saved. This property will be seen when the job is in the phase **lock**, **purge** or **expired**.

SAFETIMEDAT

this is the human readable format of **SAVETIMESTAMP**.

SAVESTATE

state of the **save** action. This property will be seen when the job is in the phase **lock**, **purge** or **expired**.

PURGETIMESTAMP

timestamp (seconds since the epoch) when the data has been purged. This property will be seen when the job is in the phase **expired**.

If the **PURGESTATE** is *RETRY*, this property will also be seen when the job is in the phase **purge**.

PURGETIMEDAT

this is the human readable format of **PURGETIMESTAMP**.

If the **PURGESTATE** is *RETRY*, this property will also be seen when the job is in the phase **purge**.

PURGESTATE

state of the **purge** action. This property will normally be seen when the job is in the phase **expired**.

If the **PURGESTATE** is *RETRY*, this property will also be seen when the job is in the phase **purge**. The *RETRY* **PURGESTATE** shows, that the **purge** action was not completely successful. The purging of the data of a job with this condition will be repeated in subsequent calls of **lots** until it succeeds.

DATA SAVE INFORMATION:**SAVE_BASEDIR**

to this directory (on the long term storage) the data of the related job is saved to. In normal cases only one directory is listed here, but in special cases there might be listed multiple directories. This property will only have content when the job is in the phase **lock** or **purge**.

FILENAME

filename and path of the saved source file. This property will only have content when data to be saved as defined in **datalist.dat** has been found on the system when collecting the data.

SAVE_SUFFIX

suffix of the saved file. This suffix correlates to the chosen compression method as printed in the **COMPRESSION** property.

A file to be restored can therefore be accessed concatenating the properties **SAVE_BASEDIR** + **FILENAME** + **SAVE_SUFFIX**.

SIZE

size of the source file in bytes.

USER

file owner name of the source file. The numeric user ID is stored with the file.

GROUP

file group name of the source file. The numeric group ID is stored with the file.

PERMISSIONS

symbolic representation of the source file's permissions as displayed by the **ls(1)** command.

PERM numeric representation of the source file's permissions which can be used with the **chmod(1)** command.

MTIME modification time of the source file in military format.

print_log [**-f** *from_date*] [**-t** *to_date*]

print the master log between the *from_date* and the *to_date*.

If the *from_date* is not specified, the log file is printed from the beginning to the *to_date*. If the *to_date* is not entered, the log file is printed from the *from_date* beginning to the end. If neither of the two dates is specified the whole log is printed.

print_logtail

display a continuous master log output.

print_session -s *sessionname*

print the session log output. Each output of **lots** is saved and due to compliance reasons never deleted. To evaluate the session name related to a certain action or a sequence of actions, use **list_action**, **list_collect**, **list_save**, **list_lock**, **list_purge**, **list_clear** or **print_log**.

-f *from_date*

begin date in the military format **YYYY-MM-DD**. To compute dates in this format, see: **input(3)**, **seconds(3)**, **timer(1)**, **today(3)**, **tomorrow(3)**, **yesterday(3)**.

-t *to_date*

end date in the military format **YYYY-MM-DD**. To compute dates in this format, see: **input(3)**, **seconds(3)**, **timer(1)**, **today(3)**, **tomorrow(3)**, **yesterday(3)**.

-j *jobname*

name of a **lots** job in the format **YYYY-MM-DD_hh.mm.ss**.

-s *sessionname*

name of a **lots** session in the format **YYYY-MM-DD_hh.mm.ss**.

-d *datalist*

name of a datalist as specified in **datalist.dat** and **schedule.dat**.

ENVIRONMENT**\$LOTS_CONFIGFILE**

configuration file of **lots**. The **-c** *configfile* command line option has preference. If the configuration file specified in **\$LOTS_CONFIGFILE** does not exist, the default configuration file **edrc/etc/lots.cfg** is read.

\$LOTS_IDENTITIES

comma separated list of identities (=hostnames) of the **lots** command. The **-i identitylist** option has preference.

EXIT STATUS

- 0** no error.
- 1** configuration file does not exist.
- 2** session could not be created. If you get this error, check the file systems where the **VARDIR/log** directory resides. See also **lots.cfg(4)**.
- 4** Usage printed.
- 5** **lots** aborted pressing <Ctrl>+<C>.
- 6** job as specified in **-j jobname** not found.
- 7** session as specified in **-s sessionname** not found.
- 8** cannot write to logfile.
- 9** cannot write to **VARDIR**.
- 10** a job to be cleared does not exist or is not in the **save** or **lock** phases.

FILES

The **VARDIR** can be defined in the **lots.cfg** file. Default is **edrc/var/lots**.

edrc/etc/lots.cfg

default **lots** configuration file.

VARDIR/log/lots.master.log

logfile of **lots**.

Do not modify or shorten this file.

VARDIR/log/lots.session.<SESSIONNAME>.log.gz

session logfile of **lots**. Display session logs using the ' **lots -a print_session -s sessionname** ' command. Do not modify or delete any of this files.

VARDIR/locks/

lockfiles, do not edit them by hand.

VARDIR/objects/datalist.dat

data save definition. In this file it is defined which set of files and directories are saved using one handle (datalist).

VARDIR/objects/schedule.dat

schedule, retention and compression definition. In this file it is defined which datalist is scheduled to be saved on which date.

VARDIR/objects/volume.dat

definition of data save volume destinations. With this file it is possible to locate long term data saves to different destinations.

VARDIR/spool/save/<JOBNAME>.job

jobs in the **save** phase. Do not access job files directly, always use the **-a print_job** action to query job information, due to the fact that certain job information is resolved by **lots** while printing the job and some job properties are constructed for certain job VERSIONs due to functionality enhancements of **lots**.

VARDIR/spool/lock/<JOBNAME>.job

jobs in the **lock** phase. Do not access job files directly, always use the **-a print_job** action to query job information, due to the fact that certain job information is resolved by **lots** while printing the job and some job properties are constructed for certain job VERSIONs due to functionality enhancements of **lots**.

VARDIR/spool/purge/<JOBNAME>.job

jobs in the **purge** phase. Do not access job files directly, always use the **-a print_job** action to query job information, due to the fact that certain job information is resolved by **lots** while printing the job and some job properties are constructed for certain job VERSIONs due to functionality enhancements of **lots**.

VARDIR/spool/expired/<JOBNAME>.job

jobs in the **expired** phase. Do not access job files directly, always use the **-a print_job** action to query job information, due to the fact that certain job information is resolved by **lots** while printing the job and some job properties are constructed for certain job VERSIONs due to functionality enhancements of **lots**.

VARDIR/state/action

record of all performed actions. Do not modify or shorten this file.

VARDIR/state/diskusage

record of used disk space. This information is used to create reports.

VARDIR/state/performance

record of durations and data throughput. This information is used to calculate forecasts and create performance reports.

<VOLUME_PATH>/<YEAR>/<MMDD>/<DATALIST>/<COUNTER>/data/

SAVE_BASEDIR directory on the long term storage device where the data as defined in a datalist is saved to.

<VOLUME_PATH>/<YEAR>/<MMDD>/<DATALIST>/<COUNTER>/meta/<JOBNAME>.job
 copy of the job file as located in **VARDIR/spool/lock/<JOBNAME>.job** or **VARDIR/spool/purge/<JOBNAME>.job** depending on the phase where the job currently is in. On a severe emergency situation where the information in the **VARDIR/spool/lock/** and/or **VARDIR/spool/purge/** directories is lost without having a backup, the state of the **lots** jobs in the **lock** and **purge** phases, which are the most critical ones, can be reconstructed by copying the job files from **<VOLUME_PATH>/<YEAR>/<MMDD>/<DATALIST>/<COUNTER>/meta/** back to the **VARDIR/spool/lock/** or **VARDIR/spool/purge/** directories. The meta data of saved data is also locked with the identical retention as the data, therefore this information is secured.

EXAMPLES

-

SEE ALSO

edrcintro(1), lots.cfg(4), datalist.dat(4), schedule.dat(4), volume.dat(4), bzip2(1), chmod(1), chown(1), compress(1), cpio(1), gzip(1), input(3), ls(1), seconds(3), timer(1), timezone(3), today(3), tomorrow(3), unzip(1), yesterday(3), zip(1).

NOTES

The NetAPP filers are able to provide the following SnapLock variants, based on the licenses applied:

SnapLock Enterprise:

Is a trusted administrator mode. In this mode a volume containing non-expired locked data can be destroyed by an NetAPP administrator on the NetAPP filer.

SnapLock Compliance:

Is a untrusted administrator mode. In this mode a volume containing non-expired locked data can not be destroyed on the NetAPP filer. This results in the risk, that on program malfunction or handling errors on the SnapLock volume, a big amount of disk space could be wasted without the possibility to clean up the data and to free up the wasted disk space.

BUGS

Abort handling is fully tested under HP-11ia only. Therefore when running **lots** on other operating systems, refrain from aborting a running **lots** session if possible. However, the expected side effects are not severe.

AUTHOR

lots was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lots.cfg – configuration file for lots

SYNOPSIS

edrc/etc/lots.cfg
\$LOTS_CONFIGFILE

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **lots** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the **=** and the **VALUE** are no spaces.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a **VALUE**.

OPTIONS

VARDIR location of the **edrc/var/lots** directory containing all state and log information. When the **VARDIR** is changed from the default location it has to be ensured that the whole structure of the **VARDIR** is present at the altered location. Ensure that the directory permissions are identical to the ones as set at the default location.

See also description in **lots(1m)** concerning this topic.

Example: **VARDIR=/dat/worm_all/var/lots**

Default: **VARDIR=edrc/var/lots**

LOG Log output dir of lots. If you specify a relative path name the path is relative to the **VARDIR** setting.

Example: **LOG=/var/opt/lots/log**

Default: **LOG=\$VARDIR/log**

DATA_LOCK_LAG

Number of days after the lock is carried out on the copied data. This is to allow corrective intervention prior to the data locking.

If the **DATA_LOCK_LAG** is bigger then the **RETENTION_MAX** setting, the effective **DATA_LOCK_LAG** is adjusted to the value of **RETENTION_MAX**.

See also description in **lots**(1m) concerning this topic.

Example: **DATA_LOCK_LAG=7**

Default: **DATA_LOCK_LAG=30**

DATA_PURGE_LAG

Number of days by which the removal of data to be purged is delayed. This is to allow smooth data purge if the operating system clocks and the compliance clocks on the storage devices are not completely in sync.

See also description in **lots**(1m) concerning this topic.

Example: **DATA_PURGE_LAG=7**

Default: **DATA_PURGE_LAG=0**

RETENTION_MIN

Minimum retention in days. Specified retentions in **schedule.dat** that are lower then this watermark will be adjusted automatically to this minimum.

Example: **RETENTION_MIN=150**

Default: **RETENTION_MIN=1**

RETENTION_MAX

Maximum retention in days. Specified retentions in **schedule.dat** that are higher then this watermark will be adjusted automatically to this maximum.

Example: **RETENTION_MAX=1460**

Default: **RETENTION_MAX=60**

LOCK_TIMEOUT

timeout in seconds. This is the maximum lifetime of a job-, datalist and session lock. This setting has to be adjusted only, if the data processing duration exceeds the default.

The **LOCK_TIMEOUT** should not exceed the minimum scheduling interval of any datalist defined.

Important: Do not confound this setting with the settings concerning the retention and the locking of saved data.

Example: LOCK_TIMEOUT=82800

Default: LOCK_TIMEOUT=79200

FORECAST_DATAPOINTS

Number of most recent data points that should be used to calculate the various forecasts. If this setting is set to 0 all available data points are considered.

Example: FORECAST_DATAPOINTS=100

Default: FORECAST_DATAPOINTS=0

SEE ALSO

edrcintro(1), **lots(1m)**, **datalist.dat(4)**, **schedule.dat(4)**, **volume.dat(4)**

NOTES

-

BUGS

-

AUTHOR

lots.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2012 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lotsctl – start edrc with a special lots (longterm data save) control configuration

SYNOPSIS

edrc/bin/lotsctl [**-h** | **-V**]

lotsctl [**-s**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Short start of **edrc** with an other configuration which points to an own script tree. Internally **lotsctl** calls **edrc -c edrc.lotsctl.cfg -n LOTSCTL_@ID@** .

lotsctl stands for " 'L**O**ng**T**erm data **S**ave **C**on**T**ro**L**' " .

The session name (as shown in the menu) is automatically set to **LOTSCTL_<id>** . Where **<id>** is the process id of the started **edrc** instance if not already a session with the same name exists, if so the **<id>** is set to a random number.

If additional short starts are needed, create a symlink from the new short start command to **sat** (see section **EXAMPLES**).

OPTIONS

-h usage message.

-V print version and patch level of **edrc** . For an explanation of the release numbering system see **edrcrevision(1)**.

-s silent startup. Startup without showing the EDRC banner.

-t no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

ENVIRONMENT

-

EXIT STATUS

see **edrc(1m)** .

FILES**etc/edrc.lotsctl.cfg**

configuration file of **lotsctl**, see **edrc.cfg(4)** for more information.

etc/lotsctl.cfg

global configuration for the lotsctl menu tree.

scripts/lotsctl/_env

root env file for the lotsctl menu tree. Use the **env** edrc command to edit the env file.

Other files see section **FILES** in **edrc(1m)** .

EXAMPLES**1) create a new short start**

This creates a new short start command **new_shortstart** that will load the configuration file **edrc.new_shortstart.cfg** . The session name will automatically be set to **NEW_SHORTSTART_<id>** .

```
[ /root ]
[ root@rh7mzv7t001 ] [bash]: cd ~edrc/bin

[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ln -s sat new_shortstart
```

SEE ALSO

sat(1), **edrc(1m)**, **edrc.cfg(4)**, **edrcintro(1)**, **edrcrevision(1)**, **lots(1m)**, **lotsctl.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

lotsctl was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2012 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

logctl.cfg – configuration file for logctl

SYNOPSIS

edrc/etc/logctl.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **logctl** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

The "Example PROD:" represents a possible productive configuration, where the directory **/dat/worm_all** is located on a SnapLock drive. The "Example EDRC:" is an installation, which can be used for test purposes. All original directories are used and are located within WA2L/edrc

LOTS_HOME

This setting points to the lots software installation directory. This setting allows to point to an alternate base directory for testing purposes. Therefore for normal operations, this should be set to the installation directory of WA2L/edrc.

Example PROD: LOTS_HOME="‘aproot’"

Example EDRC: LOTS_HOME="‘aproot’"

Default: LOTS_HOME="‘aproot’"

LOTS_VAR

Var directory of lots. This setting should point to the long term data storage to ensure independence from the server(s) where lots runs. This to avoid state information loss on possible

server re-installations and to be able to run lots from multiple servers against the same long term data storage.

Example PROD: `LOTS_VAR="/dat/worm_all/var/lots"`

Example EDRC: `LOTS_VAR="$LOTS_HOME/var/lots"`

Default: `LOTS_VAR="`aproot`/var/lots"`

LOTS_CONFIGFILE

Active configuration file of lots. This file should also be located on the long term storage due to the same reasons as the **LOTS_VAR** setting.

Example PROD: `LOTS_CONFIGFILE="/dat/worm_all/etc/lots.cfg"`

Example EDRC: `LOTS_CONFIGFILE="$LOTS_HOME/etc/lots.cfg"`

Default: `LOTS_CONFIGFILE="`aproot`/etc/lots.cfg"`

GLUP_CONFIGFILE

The **glup.cfg** file defines settings for the billing report that is created on a monthly basis based on the number of computed jobs. This setting only affects the 'stat -> rb' menu point.

Example PROD: `GLUP_CONFIGFILE="/dat/worm_all/etc/glup.cfg"`

Default EDRC: -

Default: -

REPORT_BASEDIR

Directory to save specific reports generated in the 'stats' menu.

Example PROD: `REPORT_BASEDIR="/dat/report/`server_environment`-C"`

Example EDRC: -

Default: `REPORT_BASEDIR="/var/report/`server_environment`-C"`

TRUNK_DEFAULT

Default trunk target for the 'colo' menu. For a description of the trunk default target syntax, see trunk description in **edrc(1m)**.

Example PROD: `TRUNK_DEFAULT="adm_ora1_tst"`

Example EDRC: `TRUNK_DEFAULT="`hostlist` | awk '{print $1}'"`

Default: `TRUNK_DEFAULT=""`

TRUNK_TARGETS

Trunk targets for the 'colo' menu. For a description of the trunk target syntax, see trunk description in **edrc**(1m).

Example PROD:

```
TRUNK_TARGETS="
    adm_oral_tst%-%MSSQL%saves:adm_oral_tst
    rcat_oral_tst%-%Oracle%saves:rcat_oral_tst
    asys_app1_tst%-%Misc%saves:asys_app1_tst
"
```

Example EDRC: TRUNK_TARGETS="'hostlist' 'hostname'"

Default: TRUNK_TARGETS=""

SEE ALSO

edrc(1m), **edrcintro**(1), **lotsctl**(1), **lots**(1m)

NOTES

-

BUGS

-

AUTHOR

lotsctl.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2012 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lscomp – list the files contained in a compressed archive file

SYNOPSIS

edrc/bin/lscomp [-h]

lscomp *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **lscomp** you can list the files included in a compressed archive without decompressing the whole archive file. To view a more detailed filelist you should use the **llcomp** command.

OPTIONS

-h usage message

file filename of the compressed archive file. This file is a file with the suffixes **.tgz**, **.tar.gz**, **.tar.bz**, **.tar.bz2**, **.tar.Z**, **.tar.zip**, **.cpio**, **.cpio.gz**, **.cpio.bz**, **.cpio.bz2**, **.cpio.xz**, **.cpio.Z**, **.cpio.zip**, **.depot.gz**, **.depot**, **.epub**, **.zip**, **.jar**, **.whl**, **.rock**, **.rpm**, **.deb** or **.bw**.

EXIT STATUS

- | | |
|----------|--|
| 1 | specified archive file not found. |
| 2 | file format of specified archive file not recognized. The file format is determined by the file suffix only. |
| 4 | usage displayed. |
| 6 | error while decompressing a compressed archive. This error occurs if the file is not compressed with the command that is normally assigned with the file suffix. For instance: the file my_archive.tar.gz was not really compressed with gzip or the file is corrupt. |
| 7 | error while accessing file list in archive. This error occurs if the file is not archived with the command that is normally assigned to the file suffix. For instance: the file my_archive.tar.gz was not really a compressed tar file or the tar archive is corrupt. |

SEE ALSO

edrcintro(1), **llcomp(1)**, **catcomp(1)**, **ar(1)**, **bzip2(1)**, **bwcreate(1)**, **bwcompare(1)**, **dpkg(1)**, **zip(1)**, **unzip(1)**, **compress(1)**, **cpio(1)**, **rpm(1)**, **jar(1)**, **tar(1)**, **gzip(1)**, **xz(1)**

BUGS

-

AUTHOR

lscomp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lscp – print list with cp-commands in the current working directory

SYNOPSIS

edrc/bin/lscp [**ls_options**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Print a list of **cp** commands with files in the current working directory to stdout. This is useful if many filenames have to be copied to different filenames.

The common usage is to invoke **lscp**, redirect the output to a temporary file, load the temporary file into **vi**, modify the filenames with the editing functionality of **vi** and then execute the commands in the temporary file. With this procedure it is very effective to copy many files to different filenames.

This functionality might be used often when integrating a new application into a disaster recovery solution thru copying an existing application recovery sequence. In this case many files have to be copied, but the content might be identical in many cases.

OPTIONS

ls_options all native **ls** options.

EXIT STATUS

x the exit status of **cp**.

EXAMPLES

1) common

lscp usage:

```
lscp > a; vi a; sh a; rm a
```

SEE ALSO

edrcintro(1), **ls(1)**, **lsmv(1)**, **cp(1)**

BUGS

-

AUTHOR

lscp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lsmv – print list with mv-commands in the current working directory

SYNOPSIS

edrc/bin/lsmv [**ls_options**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Print a list of **mv** commands with files in the current working directory to stdout. This is useful if many filenames have to be adjusted.

The common usage is to invoke **lsmv**, redirect the output to a temporary file, load the temporary file into **vi**, modify the filenames with the editing functionality of **vi** and then execute the commands in the temporary file. With this procedure it is very effective to rename many files.

This functionality might be used often when integrating a new application into a disaster recovery solution thru copying an existing application recovery sequence. In this case many files have to be renamed, but the content might be identical in many cases.

OPTIONS

ls_options all native **ls** options.

EXIT STATUS

x the exit status of **mv**.

EXAMPLES

1) common

lsmv usage:

```
lsmv > a; vi a; sh a; rm a
```

SEE ALSO

edrcintro(1), **ls(1)**, **lsdp(1)**, **mv(1)**

BUGS

-

AUTHOR

lsmv was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lsOf – list open files

SYNOPSIS

```
lsOf [ -?abChKlnNOPRtUvVX ] [ -A A ] [ -c c ] [ +c c ] [ +|-d d ] [ +|-D D ] [ +|-e s ] [ +|-E ] [ +|-f
[cfgGn] ] [ -F [ff] ] [ -g [s] ] [ -i [ij] ] [ -k k ] [ +|-L [ll] ] [ +|-m m ] [ +|-M ] [ -o [o] ] [ -p s ] [ +|-r
[t[m<fnt>]] ] [ -s [p:s] ] [ -S [t] ] [ -T [t] ] [ -u s ] [ +|-w ] [ -x [fl] ] [ -z [z] ] [ -Z [Z] ] [ -- ] [names]
```

DESCRIPTION

LsOf revision 4.88 lists on its standard output file information about files opened by processes for the following UNIX dialects:

Apple Darwin 9 and Mac OS X 10.[567]
 FreeBSD 8.[234], 9.0, 10.0 and 11.0 for AMD64-based systems
 Linux 2.1.72 and above for x86-based systems
 Solaris 9, 10 and 11

(See the **DISTRIBUTION** section of this manual page for information on how to obtain the latest *lsOf* revision.)

An open file may be a regular file, a directory, a block special file, a character special file, an executing text reference, a library, a stream or a network file (Internet socket, NFS file or UNIX domain socket.) A specific file or all the files in a file system may be selected by path.

Instead of a formatted display, *lsOf* will produce output that can be parsed by other programs. See the **-F**, option description, and the **OUTPUT FOR OTHER PROGRAMS** section for more information.

In addition to producing a single output list, *lsOf* will run in repeat mode. In repeat mode it will produce output, delay, then repeat the output operation until stopped with an interrupt or quit signal. See the **+|-r** [*t*[*m*<*fnt*>]] option description for more information.

OPTIONS

In the absence of any options, *lsOf* lists all open files belonging to all active processes.

If any list request option is specified, other list requests must be specifically requested – e.g., if **-U** is specified for the listing of UNIX socket files, NFS files won't be listed unless **-N** is also specified; or if a user list is specified with the **-u** option, UNIX domain socket files, belonging to users not in the list, won't be listed unless the **-U** option is also specified.

Normally list options that are specifically stated are ORed – i.e., specifying the **-i** option without an address and the **-ufoo** option produces a listing of all network files OR files belonging to processes owned by user “foo”. The exceptions are:

- 1) the ‘^’ (negated) login name or user ID (UID), specified with the **-u** option;
- 2) the ‘^’ (negated) process ID (PID), specified with the **-p** option;
- 3) the ‘^’ (negated) process group ID (PGID), specified with the **-g** option;
- 4) the ‘^’ (negated) command, specified with the **-c** option;
- 5) the ‘^’ negated TCP or UDP protocol state names, specified with the **-s** [*p:s*] option.

Since they represent exclusions, they are applied without ORing or ANDing and take effect before any other selection criteria are applied.

The **-a** option may be used to AND the selections. For example, specifying **-a**, **-U**, and **-ufoo** produces a listing of only UNIX socket files that belong to processes owned by user “foo”.

Caution: the **-a** option causes all list selection options to be ANDed; it can't be used to cause ANDing of

selected pairs of selection options by placing it between them, even though its placement there is acceptable. Wherever **-a** is placed, it causes the ANDing of all selection options.

Items of the same selection set – command names, file descriptors, network addresses, process identifiers, user identifiers, zone names, security contexts – are joined in a single ORed set and applied before the result participates in ANDing. Thus, for example, specifying **-i@aaa.bbb**, **-i@ccc.ddd**, **-a**, and **-ufff,ggg** will select the listing of files that belong to either login “fff” OR “ggg” AND have network connections to either host aaa.bbb OR ccc.ddd.

Options may be grouped together following a single prefix -- e.g., the option set “**-a -b -C**” may be stated as **-abC**. However, since values are optional following **+-f**, **-F**, **-g**, **-i**, **+-L**, **-o**, **+-r**, **-s**, **-S**, **-T**, **-x** and **-z**, when you have no values for them be careful that the following character isn’t ambiguous. For example, **-Fn** might represent the **-F** and **-n** options, or it might represent the **n** field identifier character following the **-F** option. When ambiguity is possible, start a new option with a ‘-’ character – e.g., “**-F -n**”. If the next option is a file name, follow the possibly ambiguous option with “**--**” – e.g., “**-F -- name**”.

Either the ‘+’ or the ‘-’ prefix may be applied to a group of options. Options that don’t take on separate meanings for each prefix – e.g., **-i** – may be grouped under either prefix. Thus, for example, “**+M -i**” may be stated as “**+Mi**” and the group means the same as the separate options. Be careful of prefix grouping when one or more options in the group does take on separate meanings under different prefixes – e.g., **+-M**; “**-iM**” is not the same request as “**-i +M**”. When in doubt, use separate options with appropriate prefixes.

-? -h These two equivalent options select a usage (help) output list. *Lsof* displays a shortened form of this output when it detects an error in the options supplied to it, after it has displayed messages explaining each error. (Escape the ‘?’ character as your shell requires.)

-a causes list selection options to be ANDed, as described above.

-A A is available on systems configured for AFS whose AFS kernel code is implemented via dynamic modules. It allows the *lsof* user to specify *A* as an alternate name list file where the kernel addresses of the dynamic modules might be found. See the *lsof* FAQ (The **FAQ** section gives its location.) for more information about dynamic modules, their symbols, and how they affect *lsof*.

-b causes *lsof* to avoid kernel functions that might block – *lstat(2)*, *readlink(2)*, and *stat(2)*.

See the **BLOCKS AND TIMEOUTS** and **AVOIDING KERNEL BLOCKS** sections for information on using this option.

-c c selects the listing of files for processes executing the command that begins with the characters of *c*. Multiple commands may be specified, using multiple **-c** options. They are joined in a single ORed set before participating in AND option selection.

If *c* begins with a ‘^’, then the following characters specify a command name whose processes are to be ignored (excluded.)

If *c* begins and ends with a slash (/), the characters between the slashes are interpreted as a regular expression. Shell meta-characters in the regular expression must be quoted to prevent their interpretation by the shell. The closing slash may be followed by these modifiers:

| | |
|---|--|
| b | the regular expression is a basic one. |
| i | ignore the case of letters. |
| x | the regular expression is an extended one (default). |

See the *lsof* FAQ (The **FAQ** section gives its location.) for more information on basic and extended regular expressions.

The simple command specification is tested first. If that test fails, the command regular expression is applied. If the simple command test succeeds, the command regular expression test isn’t made. This may result in “no command found for regex:” messages when *lsof*’s **-V** option is

specified.

+c *w* defines the maximum number of initial characters of the name, supplied by the UNIX dialect, of the UNIX command associated with a process to be printed in the COMMAND column. (The *lsOf* default is nine.)

Note that many UNIX dialects do not supply all command name characters to *lsOf* in the files and structures from which *lsOf* obtains command name. Often dialects limit the number of characters supplied in those sources. For example, Linux 2.4.27 and Solaris 9 both limit command name length to 16 characters.

If *w* is zero ('0'), all command characters supplied to *lsOf* by the UNIX dialect will be printed.

If *w* is less than the length of the column title, "COMMAND", it will be raised to that length.

-C disables the reporting of any path name components from the kernel's name cache. See the **KERNEL NAME CACHE** section for more information.

+d *s* causes *lsOf* to search for all open instances of directory *s* and the files and directories it contains at its top level. **+d** does NOT descend the directory tree, rooted at *s*. The **+D** *D* option may be used to request a full-descent directory tree search, rooted at directory *D*.

Processing of the **+d** option does not follow symbolic links within *s* unless the **-x** or **-x l** option is also specified. Nor does it search for open files on file system mount points on subdirectories of *s* unless the **-x** or **-x f** option is also specified.

Note: the authority of the user of this option limits it to searching for files that the user has permission to examine with the system *stat(2)* function.

-d *s* specifies a list of file descriptors (FDs) to exclude from or include in the output listing. The file descriptors are specified in the comma-separated set *s* – e.g., "cwd,1,3", "^6,^2". (There should be no spaces in the set.)

The list is an exclusion list if all entries of the set begin with '^'. It is an inclusion list if no entry begins with '^'. Mixed lists are not permitted.

A file descriptor number range may be in the set as long as neither member is empty, both members are numbers, and the ending member is larger than the starting one – e.g., "0-7" or "3-10". Ranges may be specified for exclusion if they have the '^' prefix – e.g., "^0-7" excludes all file descriptors 0 through 7.

Multiple file descriptor numbers are joined in a single ORed set before participating in AND option selection.

When there are exclusion and inclusion members in the set, *lsOf* reports them as errors and exits with a non-zero return code.

See the description of File Descriptor (FD) output values in the **OUTPUT** section for more information on file descriptor names.

+D *D* causes *lsOf* to search for all open instances of directory *D* and all the files and directories it contains to its complete depth.

Processing of the **+D** option does not follow symbolic links within *D* unless the **-x** or **-x l** option is also specified. Nor does it search for open files on file system mount points on subdirectories of *D* unless the **-x** or **-x f** option is also specified.

Note: the authority of the user of this option limits it to searching for files that the user has permission to examine with the system *stat(2)* function.

Further note: *lsOf* may process this option slowly and require a large amount of dynamic memory to do it. This is because it must descend the entire directory tree, rooted at *D*, calling *stat(2)* for each file and directory, building a list of all the files it finds, and searching that list for a match with every open file. When directory *D* is large, these steps can take a long time, so use this option prudently.

-D D directs *lsOf*'s use of the device cache file. The use of this option is sometimes restricted. See the **DEVICE CACHE FILE** section and the sections that follow it for more information on this option.

-D must be followed by a function letter; the function letter may optionally be followed by a path name. *Lsof* recognizes these function letters:

- ?** – report device cache file paths
- b** – build the device cache file
- i** – ignore the device cache file
- r** – read the device cache file
- u** – read and update the device cache file

The **b**, **r**, and **u** functions, accompanied by a path name, are sometimes restricted. When these functions are restricted, they will not appear in the description of the **-D** option that accompanies **-h** or **-?** option output. See the **DEVICE CACHE FILE** section and the sections that follow it for more information on these functions and when they're restricted.

The **?** function reports the read-only and write paths that *lsOf* can use for the device cache file, the names of any environment variables whose values *lsOf* will examine when forming the device cache file path, and the format for the personal device cache file path. (Escape the '?' character as your shell requires.)

When available, the **b**, **r**, and **u** functions may be followed by the device cache file's path. The standard default is *.lsOf_hostname* in the home directory of the real user ID that executes *lsOf*, but this could have been changed when *lsOf* was configured and compiled. (The output of the **-h** and **-?** options show the current default prefix – e.g., “.lsOf”). The suffix, *hostname*, is the first component of the host's name returned by *gethostname(2)*.

When available, the **b** function directs *lsOf* to build a new device cache file at the default or specified path.

The **i** function directs *lsOf* to ignore the default device cache file and obtain its information about devices via direct calls to the kernel.

The **r** function directs *lsOf* to read the device cache at the default or specified path, but prevents it from creating a new device cache file when none exists or the existing one is improperly structured. The **r** function, when specified without a path name, prevents *lsOf* from updating an incorrect or outdated device cache file, or creating a new one in its place. The **r** function is always available when it is specified without a path name argument; it may be restricted by the permissions of the *lsOf* process.

When available, the **u** function directs *lsOf* to read the device cache file at the default or specified path, if possible, and to rebuild it, if necessary. This is the default device cache file function when no **-D** option has been specified.

+|-e s exempts the file system whose path name is *s* from being subjected to kernel function calls that might block. The **+e** option exempts *stat(2)*, *lstat(2)* and most *readlink(2)* kernel function calls. The **-e** option exempts only *stat(2)* and *lstat(2)* kernel function calls. Multiple file systems may be specified with separate **+|-e** specifications and each may have *readlink(2)* calls exempted or not.

This option is currently implemented only for Linux.

CAUTION: this option can easily be mis-applied to other than the file system of interest, because it uses path name rather than the more reliable device and inode numbers. (Device and inode numbers are acquired via the potentially blocking *stat(2)* kernel call and are thus not available, but see the **+|-m m** option as a possible alternative way to supply device numbers.) **Use this option with great care and fully specify the path name of the file system to be exempted.**

When open files on exempted file systems are reported, it may not be possible to obtain all their information. Therefore, some information columns will be blank, the characters “UNKN” preface the values in the TYPE column, and the applicable exemption option is added in parentheses to the end of the NAME column. (Some device number information might be made available via the **+|-m m** option.)

+|-E **+E** specifies that Linux files should be displayed with endpoint information and the files of the endpoints should also be displayed.

Endpoint information is displayed in the NAME column in the form “*PID,cmd,FDmode*”. *PID* is the endpoint process ID; *cmd* is the endpoint process command; *FD* is the endpoint file’s descriptor; and *mode* is the endpoint file’s access mode. Multiple occurrences of this information can appear in a file’s NAME column.

-E specifies that Linux pipe files should only be displayed with endpoint information.

+|-f [cfgGn]

f by itself clarifies how path name arguments are to be interpreted. When followed by **c**, **f**, **g**, **G**, or **n** in any combination it specifies that the listing of kernel file structure information is to be enabled (‘+’) or inhibited (‘-’).

Normally a path name argument is taken to be a file system name if it matches a mounted-on directory name reported by *mount*(8), or if it represents a block device, named in the *mount* output and associated with a mounted directory name. When **+f** is specified, all path name arguments will be taken to be file system names, and *lsOf* will complain if any are not. This can be useful, for example, when the file system name (mounted-on device) isn’t a block device. This happens for some CD-ROM file systems.

When **-f** is specified by itself, all path name arguments will be taken to be simple files. Thus, for example, the “**-f -- /**” arguments direct *lsOf* to search for open files with a ‘/’ path name, not all open files in the ‘/’ (root) file system.

Be careful to make sure **+f** and **-f** are properly terminated and aren’t followed by a character (e.g., of the file or file system name) that might be taken as a parameter. For example, use “**--**” after **+f** and **-f** as in these examples.

```
$ lsOf +f -- /file/system/name
$ lsOf -f -- /file/name
```

The listing of information from kernel file structures, requested with the **+f [cfgGn]** option form, is normally inhibited, and is not available in whole or part for some dialects – e.g., /proc-based Linux kernels below 2.6.22. When the prefix to **f** is a plus sign (‘+’), these characters request file structure information:

| | |
|----------|---|
| c | file structure use count (not Linux) |
| f | file structure address (not Linux) |
| g | file flag abbreviations (Linux 2.6.22 and up) |
| G | file flags in hexadecimal (Linux 2.6.22 and up) |
| n | file structure node address (not Linux) |

When the prefix is minus (‘-’) the same characters disable the listing of the indicated values.

File structure addresses, use counts, flags, and node addresses may be used to detect more readily identical files inherited by child processes and identical files in use by different processes. *lsOf* column output can be sorted by output columns holding the values and listed to identify identical file use, or *lsOf* field output can be parsed by an AWK or Perl post-filter script, or by a C program.

-Ff specifies a character list, *f*, that selects the fields to be output for processing by another program, and the character that terminates each output field. Each field to be output is specified with a single character in *f*. The field terminator defaults to NL, but may be changed to NUL (000). See the **OUTPUT FOR OTHER PROGRAMS** section for a description of the field

identification characters and the field output process.

When the field selection character list is empty, all standard fields are selected (except the raw device field, security context and zone field for compatibility reasons) and the NL field terminator is used.

When the field selection character list contains only a zero ('0'), all fields are selected (except the raw device field for compatibility reasons) and the NUL terminator character is used.

Other combinations of fields and their associated field terminator character must be set with explicit entries in *f*, as described in the **OUTPUT FOR OTHER PROGRAMS** section.

When a field selection character identifies an item *lsOf* does not normally list – e.g., PPID, selected with **-R** – specification of the field character – e.g., **“-FR”** – also selects the listing of the item.

When the field selection character list contains the single character '?', *lsOf* will display a help list of the field identification characters. (Escape the '?' character as your shell requires.)

-g [*s*] excludes or selects the listing of files for the processes whose optional process group IDentification (PGID) numbers are in the comma-separated set *s* – e.g., “123” or “123,456”. (There should be no spaces in the set.)

PGID numbers that begin with '^' (negation) represent exclusions.

Multiple PGID numbers are joined in a single ORed set before participating in AND option selection. However, PGID exclusions are applied without ORing or ANDing and take effect before other selection criteria are applied.

The **-g** option also enables the output display of PGID numbers. When specified without a PGID set that's all it does.

-i [*i*] selects the listing of files any of whose Internet address matches the address specified in *i*. If no address is specified, this option selects the listing of all Internet and x.25 (HP-UX) network files.

If **-i4** or **-i6** is specified with no following address, only files of the indicated IP version, IPv4 or IPv6, are displayed. (An IPv6 specification may be used only if the dialects supports IPv6, as indicated by “[46]” and “IPv[46]” in *lsOf*'s **-h** or **-?** output.) Sequentially specifying **-i4**, followed by **-i6** is the same as specifying **-i**, and vice-versa. Specifying **-i4**, or **-i6** after **-i** is the same as specifying **-i4** or **-i6** by itself.

Multiple addresses (up to a limit of 100) may be specified with multiple **-i** options. (A port number or service name range is counted as one address.) They are joined in a single ORed set before participating in AND option selection.

An Internet address is specified in the form (Items in square brackets are optional.):

[46][*protocol*][@*hostname* | *hostaddr*][:*service* | *port*]

where:

46 specifies the IP version, IPv4 or IPv6
that applies to the following address.
'6' may be specified only if the UNIX
dialect supports IPv6. If neither '4' nor
'6' is specified, the following address
applies to all IP versions.
protocol is a protocol name – **TCP**, **UDP**
hostname is an Internet host name. Unless a
specific IP version is specified, open
network files associated with host names
of all versions will be selected.
hostaddr is a numeric Internet IPv4 address in

dot form; or an IPv6 numeric address in colon form, enclosed in brackets, if the UNIX dialect supports IPv6. When an IP version is selected, only its numeric addresses may be specified.

service is an */etc/services* name – e.g., **smtp** – or a list of them.

port is a port number, or a list of them.

IPv6 options may be used only if the UNIX dialect supports IPv6. To see if the dialect supports IPv6, run *lsOf* and specify the **-h** or **-?** (help) option. If the displayed description of the **-i** option contains “[46]” and “IPv[46]”, IPv6 is supported.

IPv4 host names and addresses may not be specified if network file selection is limited to IPv6 with **-i 6**. IPv6 host names and addresses may not be specified if network file selection is limited to IPv4 with **-i 4**. When an open IPv4 network file’s address is mapped in an IPv6 address, the open file’s type will be IPv6, not IPv4, and its display will be selected by ‘6’, not ‘4’.

At least one address component – **4**, **6**, *protocol*, *hostname*, *hostaddr*, or *service* – must be supplied. The ‘@’ character, leading the host specification, is always required; as is the ‘:’, leading the port specification. Specify either *hostname* or *hostaddr*. Specify either *service* name list or *port* number list. If a *service* name list is specified, the *protocol* may also need to be specified if the TCP, UDP and UDPLITE port numbers for the service name are different. Use any case – lower or upper – for *protocol*.

Service names and *port* numbers may be combined in a list whose entries are separated by commas and whose numeric range entries are separated by minus signs. There may be no embedded spaces, and all service names must belong to the specified *protocol*. Since service names may contain embedded minus signs, the starting entry of a range can’t be a service name; it can be a port number, however.

Here are some sample addresses:

```
-i6 – IPv6 only
TCP:25 – TCP and port 25
@1.2.3.4 – Internet IPv4 host address 1.2.3.4
@[3ffe:1ebc::1]:1234 – Internet IPv6 host address
3ffe:1ebc::1, port 1234
UDP:who – UDP who service port
TCP@lsOf.itap:513 – TCP, port 513 and host name lsOf.itap
tcp@foo:1-10,smtp,99 – TCP, ports 1 through 10,
service name smtp, port 99, host name foo
tcp@bar:1-smtp – TCP, ports 1 through smtp, host bar
:time – either TCP, UDP or UDPLITE time service port
```

-K selects the listing of tasks (threads) of processes, on dialects where task (thread) reporting is supported. (If help output – i.e., the output of the **-h** or **-?** options – shows this option, then task (thread) reporting is supported by the dialect.)

When **-K** and **-a** are both specified on Linux, and the tasks of a main process are selected by other options, the main process will also be listed as though it were a task, but without a task ID. (See the description of the TID column in the **OUTPUT** section.)

Where the FreeBSD version supports threads, all threads will be listed with their IDs.

In general threads and tasks inherit the files of the caller, but may close some and open others, so *lsOf* always reports all the open files of threads and tasks.

-k k specifies a kernel name list file, *k*, in place of */vmunix*, */mach*, etc. **-k** is not available under AIX on the IBM RISC/System 6000.

- l** inhibits the conversion of user ID numbers to login names. It is also useful when login name lookup is working improperly or slowly.
- +|-L [*l*]** enables (+) or disables (-) the listing of file link counts, where they are available – e.g., they aren't available for sockets, or most FIFOs and pipes.
- When **+L** is specified without a following number, all link counts will be listed. When **-L** is specified (the default), no link counts will be listed.
- When **+L** is followed by a number, only files having a link count less than that number will be listed. (No number may follow **-L**.) A specification of the form “**+L1**” will select open files that have been unlinked. A specification of the form “**+aL1 <file_system>**” will select unlinked open files on the specified file system.
- For other link count comparisons, use field output (**-F**) and a post-processing script or program.
- +|-m m** specifies an alternate kernel memory file or activates mount table supplement processing.
- The option form **-m m** specifies a kernel memory file, *m*, in place of */dev/kmem* or */dev/mem* – e.g., a crash dump file.
- The option form **+m** requests that a mount supplement file be written to the standard output file. All other options are silently ignored.
- There will be a line in the mount supplement file for each mounted file system, containing the mounted file system directory, followed by a single space, followed by the device number in hexadecimal “0x” format – e.g.,
- / 0x801*
- Lsof* can use the mount supplement file to get device numbers for file systems when it can't get them via *stat(2)* or *lstat(2)*.
- The option form **+m m** identifies *m* as a mount supplement file.
- Note: the **+m** and **+m m** options are not available for all supported dialects. Check the output of *lsof*'s **-h** or **-?** options to see if the **+m** and **+m m** options are available.
- +|-M** Enables (+) or disables (-) the reporting of portmapper registrations for local TCP, UDP and UDPLITE ports, where port mapping is supported. (See the last paragraph of this option description for information about where portmapper registration reporting is supported.)
- The default reporting mode is set by the *lsof* builder with the `HASPMAPENABLED` #define in the dialect's `machine.h` header file; *lsof* is distributed with the `HASPMAPENABLED` #define deactivated, so portmapper reporting is disabled by default and must be requested with **+M**. Specifying *lsof*'s **-h** or **-?** option will report the default mode. Disabling portmapper registration when it is already disabled or enabling it when already enabled is acceptable. When portmapper registration reporting is enabled, *lsof* displays the portmapper registration (if any) for local TCP, UDP or UDPLITE ports in square brackets immediately following the port numbers or service names – e.g., “:1234[name]” or “:name[100083]”. The registration information may be a name or number, depending on what the registering program supplied to the portmapper when it registered the port.
- When portmapper registration reporting is enabled, *lsof* may run a little more slowly or even become blocked when access to the portmapper becomes congested or stopped. Reverse the reporting mode to determine if portmapper registration reporting is slowing or blocking *lsof*.
- For purposes of portmapper registration reporting *lsof* considers a TCP, UDP or UDPLITE port local if: it is found in the local part of its containing kernel structure; or if it is located in the foreign part of its containing kernel structure and the local and foreign Internet addresses are the same; or if it is located in the foreign part of its containing kernel structure and the foreign Internet address is `INADDR_LOOPBACK` (127.0.0.1). This rule may make *lsof* ignore some foreign ports on machines with multiple interfaces when the foreign Internet address is on a different interface from the local one.

See the *lsuf* FAQ (The **FAQ** section gives its location.) for further discussion of portmapper registration reporting issues.

Portmapper registration reporting is supported only on dialects that have RPC header files. (Some Linux distributions with Glibc 2.14 do not have them.) When portmapper registration reporting is supported, the **-h** or **-?** help output will show the **+|-M** option.

- n** inhibits the conversion of network numbers to host names for network files. Inhibiting conversion may make *lsuf* run faster. It is also useful when host name lookup is not working properly.
- N** selects the listing of NFS files.

- o** directs *lsuf* to display file offset at all times. It causes the SIZE/OFF output column title to be changed to OFFSET. Note: on some UNIX dialects *lsuf* can't obtain accurate or consistent file offset information from its kernel data sources, sometimes just for particular kinds of files (e.g., socket files.) Consult the *lsuf* FAQ (The **FAQ** section gives its location.) for more information.

The **-o** and **-s** options are mutually exclusive; they can't both be specified. When neither is specified, *lsuf* displays whatever value – size or offset – is appropriate and available for the type of the file.

- o o** defines the number of decimal digits (*o*) to be printed after the “0t” for a file offset before the form is switched to “0x...”. An *o* value of zero (unlimited) directs *lsuf* to use the “0t” form for all offset output.

This option does NOT direct *lsuf* to display offset at all times; specify **-o** (without a trailing number) to do that. **-o o** only specifies the number of digits after “0t” in either mixed size and offset or offset-only output. Thus, for example, to direct *lsuf* to display offset at all times with a decimal digit count of 10, use:

```
-o -o 10
```

or

```
-oo10
```

The default number of digits allowed after “0t” is normally 8, but may have been changed by the *lsuf* builder. Consult the description of the **-o o** option in the output of the **-h** or **-?** option to determine the default that is in effect.

- O** directs *lsuf* to bypass the strategy it uses to avoid being blocked by some kernel operations – i.e., doing them in forked child processes. See the **BLOCKS AND TIMEOUTS** and **AVOIDING KERNEL BLOCKS** sections for more information on kernel operations that may block *lsuf*.

While use of this option will reduce *lsuf* startup overhead, it may also cause *lsuf* to hang when the kernel doesn't respond to a function. Use this option cautiously.

- p s** excludes or selects the listing of files for the processes whose optional process IDentification (PID) numbers are in the comma-separated set *s* – e.g., “123” or “123,^456”. (There should be no spaces in the set.)

PID numbers that begin with “^” (negation) represent exclusions.

Multiple process ID numbers are joined in a single ORed set before participating in AND option selection. However, PID exclusions are applied without ORing or ANDing and take effect before other selection criteria are applied.

- P** inhibits the conversion of port numbers to port names for network files. Inhibiting the conversion may make *lsuf* run a little faster. It is also useful when port name lookup is not working properly.

- +|-r [t[m<fmt>]]**

puts *lsuf* in repeat mode. There *lsuf* lists open files as selected by other options, delays *t* seconds (default fifteen), then repeats the listing, delaying and listing repetitively until stopped by a condition defined by the prefix to the option.

If the prefix is a ‘-’, repeat mode is endless. *Lsof* must be terminated with an interrupt or quit signal.

If the prefix is ‘+’, repeat mode will end the first cycle no open files are listed – and of course when *lsof* is stopped with an interrupt or quit signal. When repeat mode ends because no files are listed, the process exit code will be zero if any open files were ever listed; one, if none were ever listed.

Lsof marks the end of each listing: if field output is in progress (the **-F**, option has been specified), the default marker is ‘m’; otherwise the default marker is “=====”. The marker is followed by a NL character.

The optional “m<fmt>” argument specifies a format for the marker line. The <fmt> characters following ‘m’ are interpreted as a format specification to the *strftime*(3) function, when both it and the *localtime*(3) function are available in the dialect’s C library. Consult the *strftime*(3) documentation for what may appear in its format specification. Note that when field output is requested with the **-F** option, <fmt> cannot contain the NL format, “%n”. Note also that when <fmt> contains spaces or other characters that affect the shell’s interpretation of arguments, <fmt> must be quoted appropriately.

Repeat mode reduces *lsof* startup overhead, so it is more efficient to use this mode than to call *lsof* repetitively from a shell script, for example.

To use repeat mode most efficiently, accompany **+|-r** with specification of other *lsof* selection options, so the amount of kernel memory access *lsof* does will be kept to a minimum. Options that filter at the process level – e.g., **-c**, **-g**, **-p**, **-u** – are the most efficient selectors.

Repeat mode is useful when coupled with field output (see the **-F**, option description) and a supervising *awk* or *Perl* script, or a C program.

-R directs *lsof* to list the Parent Process IDentification number in the PPID column.

-s [p:s] *s* alone directs *lsof* to display file size at all times. It causes the SIZE/OFF output column title to be changed to SIZE. If the file does not have a size, nothing is displayed.

The optional **-s p:s** form is available only for selected dialects, and only when the **-h** or **-?** help output lists it.

When the optional form is available, the *s* may be followed by a protocol name (*p*), either TCP or UDP, a colon (‘:’) and a comma-separated protocol state name list, the option causes open TCP and UDP files to be excluded if their state name(s) are in the list (*s*) preceded by a ‘^’; or included if their name(s) are not preceded by a ‘^’.

When an inclusion list is defined, only network files with state names in the list will be present in the *lsof* output. Thus, specifying one state name means that only network files with that lone state name will be listed.

Case is unimportant in the protocol or state names, but there may be no spaces and the colon (‘:’) separating the protocol name (*p*) and the state name list (*s*) is required.

If only TCP and UDP files are to be listed, as controlled by the specified exclusions and inclusions, the **-i** option must be specified, too. If only a single protocol’s files are to be listed, add its name as an argument to the **-i** option.

For example, to list only network files with TCP state LISTEN, use:

```
-iTCP -sTCP:LISTEN
```

Or, for example, to list network files with all UDP states except Idle, use:

```
-iUDP -sUDP:Idle
```

State names vary with UNIX dialects, so it’s not possible to provide a complete list. Some common TCP state names are: CLOSED, IDLE, BOUND, LISTEN, ESTABLISHED, SYN_SENT, SYN_RCDV, ESTABLISHED, CLOSE_WAIT, FIN_WAIT1, CLOSING, LAST_ACK,

FIN_WAIT_2, and TIME_WAIT. Two common UDP state names are Unbound and Idle.

See the *lsuf* FAQ (The **FAQ** section gives its location.) for more information on how to use protocol state exclusion and inclusion, including examples.

The **-o** (without a following decimal digit count) and **-s** option (without a following protocol and state name list) are mutually exclusive; they can't both be specified. When neither is specified, *lsuf* displays whatever value – size or offset – is appropriate and available for the type of file.

Since some types of files don't have true sizes – sockets, FIFOs, pipes, etc. – *lsuf* displays for their sizes the content amounts in their associated kernel buffers, if possible.

-S [t] specifies an optional time-out seconds value for kernel functions – *lstat(2)*, *readlink(2)*, and *stat(2)* – that might otherwise deadlock. The minimum for *t* is two; the default, fifteen; when no value is specified, the default is used.

See the **BLOCKS AND TIMEOUTS** section for more information.

-T [t] controls the reporting of some TCP/TPI information, also reported by *netstat(1)*, following the network addresses. In normal output the information appears in parentheses, each item except TCP or TPI state name identified by a keyword, followed by '=', separated from others by a single space:

```
<TCP or TPI state name>
QR=<read queue length>
QS=<send queue length>
SO=<socket options and values>
SS=<socket states>
TF=<TCP flags and values>
WR=<window read length>
WW=<window write length>
```

Not all values are reported for all UNIX dialects. Items values (when available) are reported after the item name and '='.

When the field output mode is in effect (See **OUTPUT FOR OTHER PROGRAMS**.) each item appears as a field with a 'T' leading character.

-T with no following key characters disables TCP/TPI information reporting.

-T with following characters selects the reporting of specific TCP/TPI information:

```
f      selects reporting of socket options,
        states and values, and TCP flags and
        values.
q      selects queue length reporting.
s      selects connection state reporting.
w      selects window size reporting.
```

Not all selections are enabled for some UNIX dialects. State may be selected for all dialects and is reported by default. The **-h** or **-?** help output for the **-T** option will show what selections may be used with the UNIX dialect.

When **-T** is used to select information – i.e., it is followed by one or more selection characters – the displaying of state is disabled by default, and it must be explicitly selected again in the characters following **-T**. (In effect, then, the default is equivalent to **-Ts**.) For example, if queue lengths and state are desired, use **-Tqs**.

Socket options, socket states, some socket values, TCP flags and one TCP value may be reported (when available in the UNIX dialect) in the form of the names that commonly appear after *SO_*, *so_*, *SS_*, *TCP_* and *TF_* in the dialect's header files – most often *<sys/socket.h>*, *<sys/socketvar.h>* and *<netinet/tcp_var.h>*. Consult those header files for the meaning of the

flags, options, states and values.

“SO=” precedes socket options and values; “SS=”, socket states; and “TF=”, TCP flags and values.

If a flag or option has a value, the value will follow an ‘=’ and the name -- e.g., “SO=LINGER=5”, “SO=QLIM=5”, “TF=MSS=512”. The following seven values may be reported:

| Name Reported | Description (Common Symbol) |
|------------------|--------------------------------------|
| KEEPALIVE | keep alive time (SO_KEEPALIVE) |
| LINGER | linger time (SO_LINGER) |
| MSS | maximum segment size (TCP_MAXSEG) |
| PQLEN | partial listen queue connections |
| QLEN | established listen queue connections |
| QLIM | established listen queue limit |
| RCVBUF | receive buffer length (SO_RCVBUF) |
| SNDBUF | send buffer length (SO_SNDBUF) |

Details on what socket options and values, socket states, and TCP flags and values may be displayed for particular UNIX dialects may be found in the answer to the “Why doesn’t *lsf* report socket options, socket states, and TCP flags and values for my dialect?” and “Why doesn’t *lsf* report the partial listen queue connection count for my dialect?” questions in the *lsf* FAQ (The **FAQ** section gives its location.)

-t specifies that *lsf* should produce terse output with process identifiers only and no header – e.g., so that the output may be piped to *kill*(1). **-t** selects the **-w** option.

-u s selects the listing of files for the user whose login names or user ID numbers are in the comma-separated set *s* – e.g., “abe”, or “548,root”. (There should be no spaces in the set.)

Multiple login names or user ID numbers are joined in a single ORed set before participating in AND option selection.

If a login name or user ID is preceded by a ‘^’, it becomes a negation – i.e., files of processes owned by the login name or user ID will never be listed. A negated login name or user ID selection is neither ANDed nor ORed with other selections; it is applied before all other selections and absolutely excludes the listing of the files of the process. For example, to direct *lsf* to exclude the listing of files belonging to root processes, specify “-u^root” or “-u^0”.

-U selects the listing of UNIX domain socket files.

-v selects the listing of *lsf* version information, including: revision number; when the *lsf* binary was constructed; who constructed the binary and where; the name of the compiler used to construct the *lsf* binary; the version number of the compiler when readily available; the compiler and loader flags used to construct the *lsf* binary; and system information, typically the output of *uname*’s **-a** option.

-V directs *lsf* to indicate the items it was asked to list and failed to find – command names, file names, Internet addresses or files, login names, NFS files, PIDs, PGIDs, and UIDs.

When other options are ANDed to search options, or compile-time options restrict the listing of some files, *lsf* may not report that it failed to find a search item when an ANDed option or compile-time option prevents the listing of the open file containing the located search item.

For example, “*lsf* -V -iTCP@foobar -a -d 999” may not report a failure to locate open files at “TCP@foobar” and may not list any, if none have a file descriptor number of 999. A similar situation arises when HASSECURITY and HASNOSOCKSECURITY are defined at compile time and they prevent the listing of open files.

- +|-w** Enables (+) or disables (-) the suppression of warning messages.
- The *lsOf* builder may choose to have warning messages disabled or enabled by default. The default warning message state is indicated in the output of the **-h** or **-?** option. Disabling warning messages when they are already disabled or enabling them when already enabled is acceptable.
- The **-t** option selects the **-w** option.
- x** [*fl*] may accompany the **+d** and **+D** options to direct their processing to cross over symbolic links and/or file system mount points encountered when scanning the directory (**+d**) or directory tree (**+D**).
- If **-x** is specified by itself without a following parameter, cross-over processing of both symbolic links and file system mount points is enabled. Note that when **-x** is specified without a parameter, the next argument must begin with '-' or '+'.
 The optional 'f' parameter enables file system mount point cross-over processing; 'l', symbolic link cross-over processing.
- The **-x** option may not be supplied without also supplying a **+d** or **+D** option.
- X** This is a dialect-specific option.
- AIX:**
- This IBM AIX RISC/System 6000 option requests the reporting of executed text file and shared library references.
- WARNING:** because this option uses the kernel `readx()` function, its use on a busy AIX system might cause an application process to hang so completely that it can neither be killed nor stopped. I have never seen this happen or had a report of its happening, but I think there is a remote possibility it could happen.
- By default use of `readx()` is disabled. On AIX 5L and above *lsOf* may need `setuid-root` permission to perform the actions this option requests.
- The *lsOf* builder may specify that the **-X** option be restricted to processes whose real UID is root. If that has been done, the **-X** option will not appear in the **-h** or **-?** help output unless the real UID of the *lsOf* process is root. The default *lsOf* distribution allows any UID to specify **-X**, so by default it will appear in the help output.
- When AIX `readx()` use is disabled, *lsOf* may not be able to report information for all text and loader file references, but it may also avoid exacerbating an AIX kernel directory search kernel error, known as the Stale Segment ID bug.
- The `readx()` function, used by *lsOf* or any other program to access some sections of kernel virtual memory, can trigger the Stale Segment ID bug. It can cause the kernel's `dir_search()` function to believe erroneously that part of an in-memory copy of a file system directory has been zeroed. Another application process, distinct from *lsOf*, asking the kernel to search the directory – e.g., by using `open(2)` – can cause `dir_search()` to loop forever, thus hanging the application process.
- Consult the *lsOf* FAQ (The **FAQ** section gives its location.) and the *00README* file of the *lsOf* distribution for a more complete description of the Stale Segment ID bug, its APAR, and methods for defining `readx()` use when compiling *lsOf*.
- Linux:**
- This Linux option requests that *lsOf* skip the reporting of information on all open TCP, UDP and UDPLITE IPv4 and IPv6 files.
- This Linux option is most useful when the system has an extremely large number of open TCP, UDP and UDPLITE files, the processing of whose information in the `/proc/net/tcp*` and `/proc/net/udp*` files would take *lsOf* a long time, and whose reporting is not of interest.

Use this option with care and only when you are sure that the information you want *lsOf* to display isn't associated with open TCP, UDP or UDPLITE socket files.

Solaris 10 and above:

This Solaris 10 and above option requests the reporting of cached paths for files that have been deleted – i.e., removed with *rm*(1) or *unlink*(2).

The cached path is followed by the string “ (deleted)” to indicate that the path by which the file was opened has been deleted.

Because intervening changes made to the path – i.e., renames with *mv*(1) or *rename*(2) – are not recorded in the cached path, what *lsOf* reports is only the path by which the file was opened, not its possibly different final path.

-z [z] specifies how Solaris 10 and higher zone information is to be handled.

Without a following argument – e.g., NO *z* – the option specifies that zone names are to be listed in the ZONE output column.

The **-z** option may be followed by a zone name, *z*. That causes *lsOf* to list only open files for processes in that zone. Multiple **-z z** option and argument pairs may be specified to form a list of named zones. Any open file of any process in any of the zones will be listed, subject to other conditions specified by other options and arguments.

-Z [Z] specifies how SELinux security contexts are to be handled. It and 'Z' field output character support are inhibited when SELinux is disabled in the running Linux kernel. See **OUTPUT FOR OTHER PROGRAMS** for more information on the 'Z' field output character.

Without a following argument – e.g., NO *Z* – the option specifies that security contexts are to be listed in the SECURITY-CONTEXT output column.

The **-Z** option may be followed by a wildcard security context name, *Z*. That causes *lsOf* to list only open files for processes in that security context. Multiple **-Z Z** option and argument pairs may be specified to form a list of security contexts. Any open file of any process in any of the security contexts will be listed, subject to other conditions specified by other options and arguments. Note that *Z* can be A:B:C or *:B:C or A:B:* or *:*:C to match against the A:B:C context.

-- The double minus sign option is a marker that signals the end of the keyed options. It may be used, for example, when the first file name begins with a minus sign. It may also be used when the absence of a value for the last keyed option must be signified by the presence of a minus sign in the following option and before the start of the file names.

names These are path names of specific files to list. Symbolic links are resolved before use. The first name may be separated from the preceding options with the “--” option.

If a *name* is the mounted-on directory of a file system or the device of the file system, *lsOf* will list all the files open on the file system. To be considered a file system, the *name* must match a mounted-on directory name in *mount*(8) output, or match the name of a block device associated with a mounted-on directory name. The **+|-f** option may be used to force *lsOf* to consider a *name* a file system identifier (**+f**) or a simple file (**-f**).

If *name* is a path to a directory that is not the mounted-on directory name of a file system, it is treated just as a regular file is treated – i.e., its listing is restricted to processes that have it open as a file or as a process-specific directory, such as the root or current working directory. To request that *lsOf* look for open files inside a directory name, use the **+d s** and **+D D** options.

If a *name* is the base name of a family of multiplexed files – e. g, AIX's */dev/pt[cs]* – *lsOf* will list all the associated multiplexed files on the device that are open – e.g., */dev/pt[cs]/1*, */dev/pt[cs]/2*, etc.

If a *name* is a UNIX domain socket name, *lsOf* will usually search for it by the characters of the name alone – exactly as it is specified and is recorded in the kernel socket structure. (See the

next paragraph for an exception to that rule for Linux.) Specifying a relative path – e.g., *./file* – in place of the file’s absolute path – e.g., */tmp/file* – won’t work because *lsOf* must match the characters you specify with what it finds in the kernel UNIX domain socket structures.

If a *name* is a Linux UNIX domain socket name, in one case *lsOf* is able to search for it by its device and inode number, allowing *name* to be a relative path. The case requires that the absolute path -- i.e., one beginning with a slash ('/') be used by the process that created the socket, and hence be stored in the */proc/net/unix* file; and it requires that *lsOf* be able to obtain the device and node numbers of both the absolute path in */proc/net/unix* and *name* via successful *stat(2)* system calls. When those conditions are met, *lsOf* will be able to search for the UNIX domain socket when some path to it is specified in *name*. Thus, for example, if the path is */dev/log*, and an *lsOf* search is initiated when the working directory is */dev*, then *name* could be *./log*.

If a *name* is none of the above, *lsOf* will list any open files whose device and inode match that of the specified path *name*.

If you have also specified the **-b** option, the only *names* you may safely specify are file systems for which your mount table supplies alternate device numbers. See the **AVOIDING KERNEL BLOCKS** and **ALTERNATE DEVICE NUMBERS** sections for more information.

Multiple file names are joined in a single ORed set before participating in AND option selection.

AFS

lsOf supports the recognition of AFS files for these dialects (and AFS versions):

- AIX 4.1.4 (AFS 3.4a)
- HP-UX 9.0.5 (AFS 3.4a)
- Linux 1.2.13 (AFS 3.3)
- Solaris 2.[56] (AFS 3.4a)

It may recognize AFS files on other versions of these dialects, but has not been tested there. Depending on how AFS is implemented, *lsOf* may recognize AFS files in other dialects, or may have difficulties recognizing AFS files in the supported dialects.

lsOf may have trouble identifying all aspects of AFS files in supported dialects when AFS kernel support is implemented via dynamic modules whose addresses do not appear in the kernel’s variable name list. In that case, *lsOf* may have to guess at the identity of AFS files, and might not be able to obtain volume information from the kernel that is needed for calculating AFS volume node numbers. When *lsOf* can’t compute volume node numbers, it reports blank in the NODE column.

The **-A A** option is available in some dialect implementations of *lsOf* for specifying the name list file where dynamic module kernel addresses may be found. When this option is available, it will be listed in the *lsOf* help output, presented in response to the **-h** or **-?**

See the *lsOf* FAQ (The **FAQ** section gives its location.) for more information about dynamic modules, their symbols, and how they affect *lsOf* options.

Because AFS path lookups don’t seem to participate in the kernel’s name cache operations, *lsOf* can’t identify path name components for AFS files.

SECURITY

lsOf has three features that may cause security concerns. First, its default compilation mode allows anyone to list all open files with it. Second, by default it creates a user-readable and user-writable device cache file in the home directory of the real user ID that executes *lsOf*. (The list-all-open-files and device cache features may be disabled when *lsOf* is compiled.) Third, its **-k** and **-m** options name alternate kernel name list or memory files.

Restricting the listing of all open files is controlled by the compile-time HASSECURITY and HASNOSOCKSECURITY options. When HASSECURITY is defined, *lsOf* will allow only the root user to list all open files. The non-root user may list only open files of processes with the same user IDentification

number as the real user ID number of the *lsof* process (the one that its user logged on with).

However, if `HASSECURITY` and `HASNOSOCKSECURITY` are both defined, anyone may list open socket files, provided they are selected with the `-i` option.

When `HASSECURITY` is not defined, anyone may list all open files.

Help output, presented in response to the `-h` or `-?` option, gives the status of the `HASSECURITY` and `HASNOSOCKSECURITY` definitions.

See the **Security** section of the `00README` file of the *lsof* distribution for information on building *lsof* with the `HASSECURITY` and `HASNOSOCKSECURITY` options enabled.

Creation and use of a user-readable and user-writable device cache file is controlled by the compile-time `HASDCACHE` option. See the **DEVICE CACHE FILE** section and the sections that follow it for details on how its path is formed. For security considerations it is important to note that in the default *lsof* distribution, if the real user ID under which *lsof* is executed is root, the device cache file will be written in root's home directory – e.g., `/` or `/root`. When `HASDCACHE` is not defined, *lsof* does not write or attempt to read a device cache file.

When `HASDCACHE` is defined, the *lsof* help output, presented in response to the `-h`, `-D?`, or `-?` options, will provide device cache file handling information. When `HASDCACHE` is not defined, the `-h` or `-?` output will have no `-D` option description.

Before you decide to disable the device cache file feature – enabling it improves the performance of *lsof* by reducing the startup overhead of examining all the nodes in `/dev` (or `/devices`) – read the discussion of it in the `00DCACHE` file of the *lsof* distribution and the *lsof* FAQ (The **FAQ** section gives its location.)

WHEN IN DOUBT, YOU CAN TEMPORARILY DISABLE THE USE OF THE DEVICE CACHE FILE WITH THE `-Di` OPTION.

When *lsof* user declares alternate kernel name list or memory files with the `-k` and `-m` options, *lsof* checks the user's authority to read them with `access(2)`. This is intended to prevent whatever special power *lsof*'s modes might confer on it from letting it read files not normally accessible via the authority of the real user ID.

OUTPUT

This section describes the information *lsof* lists for each open file. See the **OUTPUT FOR OTHER PROGRAMS** section for additional information on output that can be processed by another program.

Lsof only outputs printable (declared so by `isprint(3)`) 8 bit characters. Non-printable characters are printed in one of three forms: the C “`\bfrnt`” form; the control character “`^`” form (e.g., “`^@`”); or hexadecimal leading “`\x`” form (e.g., “`\xab`”). Space is non-printable in the **COMMAND** column (“`\x20`”) and printable elsewhere.

For some dialects – if `HASSETLOCALE` is defined in the dialect's machine.h header file – *lsof* will print the extended 8 bit characters of a language locale. The *lsof* process must be supplied a language locale environment variable (e.g., `LANG`) whose value represents a known language locale in which the extended characters are considered printable by `isprint(3)`. Otherwise *lsof* considers the extended characters non-printable and prints them according to its rules for non-printable characters, stated above. Consult your dialect's `setlocale(3)` man page for the names of other environment variables that may be used in place of `LANG` – e.g., `LC_ALL`, `LC_CTYPE`, etc.

Lsof's language locale support for a dialect also covers wide characters – e.g., UTF-8 – when `HASSETLOCALE` and `HASWIDECHAR` are defined in the dialect's machine.h header file, and when a suitable language locale has been defined in the appropriate environment variable for the *lsof* process. Wide characters are printable under those conditions if `iswprint(3)` reports them to be. If `HASSETLOCALE`, `HASWIDECHAR` and a suitable language locale aren't defined, or if `iswprint(3)` reports wide characters that aren't printable, *lsof* considers the wide characters non-printable and prints each of their 8 bits according to its rules for non-printable characters, stated above.

Consult the answers to the "Language locale support" questions in the *lsof* FAQ (The **FAQ** section gives its location.) for more information.

Lsof dynamically sizes the output columns each time it runs, guaranteeing that each column is a minimum size. It also guarantees that each column is separated from its predecessor by at least one space.

COMMAND contains the first nine characters of the name of the UNIX command associated with the process. If a non-zero *w* value is specified to the **+c w** option, the column contains the first *w* characters of the name of the UNIX command associated with the process up to the limit of characters supplied to *lsof* by the UNIX dialect. (See the description of the **+c w** command or the *lsof* FAQ for more information. The **FAQ** section gives its location.)

If *w* is less than the length of the column title, “COMMAND”, it will be raised to that length.

If a zero *w* value is specified to the **+c w** option, the column contains all the characters of the name of the UNIX command associated with the process.

All command name characters maintained by the kernel in its structures are displayed in field output when the command name descriptor (‘c’) is specified. See the **OUTPUT FOR OTHER COMMANDS** section for information on selecting field output and the associated command name descriptor.

PID is the Process IDentification number of the process.

TID is the task (thread) IDentification number, if task (thread) reporting is supported by the dialect and a task (thread) is being listed. (If help output – i.e., the output of the **-h** or **-?** options – shows this option, then task (thread) reporting is supported by the dialect.)

A blank TID column in Linux indicates a process – i.e., a non-task.

ZONE is the Solaris 10 and higher zone name. This column must be selected with the **-z** option.

SECURITY-CONTEXT

is the SELinux security context. This column must be selected with the **-Z** option. Note that the **-Z** option is inhibited when SELinux is disabled in the running Linux kernel.

PPID is the Parent Process IDentification number of the process. It is only displayed when the **-R** option has been specified.

PGID is the process group IDentification number associated with the process. It is only displayed when the **-g** option has been specified.

USER is the user ID number or login name of the user to whom the process belongs, usually the same as reported by *ps*(1). However, on Linux **USER** is the user ID number or login that owns the directory in */proc* where *lsof* finds information about the process. Usually that is the same value reported by *ps*(1), but may differ when the process has changed its effective user ID. (See the **-l** option description for information on when a user ID number or login name is displayed.)

FD is the File Descriptor number of the file or:

| | |
|-------------|---|
| cwd | current working directory; |
| Lnn | library references (AIX); |
| err | FD information error (see NAME column); |
| jld | jail directory (FreeBSD); |
| ltx | shared library text (code and data); |
| Mxx | hex memory-mapped type number xx. |
| m86 | DOS Merge mapped file; |
| mem | memory-mapped file; |
| mmap | memory-mapped device; |
| pd | parent directory; |
| rtd | root directory; |
| tr | kernel trace file (OpenBSD); |

txt program text (code and data);
v86 VP/ix mapped file;

FD is followed by one of these characters, describing the mode under which the file is open:

r for read access;
w for write access;
u for read and write access;
 space if mode unknown and no lock character follows;
 ‘-’ if mode unknown and lock character follows.

The mode character is followed by one of these lock characters, describing the type of lock applied to the file:

N for a Solaris NFS lock of unknown type;
r for read lock on part of the file;
R for a read lock on the entire file;
w for a write lock on part of the file;
W for a write lock on the entire file;
u for a read and write lock of any length;
U for a lock of unknown type;
x for an SCO OpenServer Xenix lock on part of the file;
X for an SCO OpenServer Xenix lock on the entire file;
 space if there is no lock.

See the **LOCKS** section for more information on the lock information character.

TYPE The FD column contents constitutes a single field for parsing in post-processing scripts. is the type of the node associated with the file – e.g., GDIR, GREG, VDIR, VREG, etc. or “IPv4” for an IPv4 socket;
 or “IPv6” for an open IPv6 network file – even if its address is IPv4, mapped in an IPv6 address;
 or “ax25” for a Linux AX.25 socket;
 or “inet” for an Internet domain socket;
 or “lla” for a HP-UX link level access file;
 or “rte” for an AF_ROUTE socket;
 or “sock” for a socket of unknown domain;
 or “unix” for a UNIX domain socket;
 or “x.25” for an HP-UX x.25 socket;
 or “BLK” for a block special file;
 or “CHR” for a character special file;
 or “DEL” for a Linux map file that has been deleted;
 or “DIR” for a directory;
 or “DOOR” for a VDOOR file;
 or “FIFO” for a FIFO special file;
 or “KQUEUE” for a BSD style kernel event queue file;
 or “LINK” for a symbolic link file;

or “MPB” for a multiplexed block file;
or “MPC” for a multiplexed character file;
or “NOFD” for a Linux `/proc/<PID>/fd` directory that can’t be opened -- the directory path appears in the NAME column, followed by an error message;
or “PAS” for a `/proc/as` file;
or “PAXV” for a `/proc/auxv` file;
or “PCRE” for a `/proc/cred` file;
or “PCTL” for a `/proc` control file;
or “PCUR” for the current `/proc` process;
or “PCWD” for a `/proc` current working directory;
or “PDIR” for a `/proc` directory;
or “PETY” for a `/proc` executable type (*etype*);
or “PFD” for a `/proc` file descriptor;
or “PFDR” for a `/proc` file descriptor directory;
or “PFIL” for an executable `/proc` file;
or “PFPR” for a `/proc` FP register set;
or “PGD” for a `/proc/pagedata` file;
or “PGID” for a `/proc` group notifier file;
or “PIPE” for pipes;
or “PLC” for a `/proc/lwpctl` file;
or “PLDR” for a `/proc/lpw` directory;
or “PLDT” for a `/proc/ldt` file;
or “PLPI” for a `/proc/lpsinfo` file;
or “PLST” for a `/proc/lstatus` file;
or “PLU” for a `/proc/lusage` file;
or “PLWG” for a `/proc/gwindows` file;
or “PLWI” for a `/proc/lwpsinfo` file;
or “PLWS” for a `/proc/lwpstatus` file;
or “PLWU” for a `/proc/lwpusage` file;
or “PLWX” for a `/proc/xregs` file;
or “PMAP” for a `/proc` map file (*map*);
or “PMEM” for a `/proc` memory image file;
or “PNTF” for a `/proc` process notifier file;
or “POBJ” for a `/proc/object` file;
or “PODR” for a `/proc/object` directory;
or “POLP” for an old format `/proc` light weight process file;
or “POPF” for an old format `/proc` PID file;
or “POPG” for an old format `/proc` page data file;
or “PORT” for a SYSV named pipe;

or “PREG” for a */proc* register file;
 or “PRMP” for a */proc/rmap* file;
 or “PRTD” for a */proc* root directory;
 or “PSGA” for a */proc/sigact* file;
 or “PSIN” for a */proc/psinfo* file;
 or “PSTA” for a */proc* status file;
 or “PSXSEM” for a POSIX semaphore file;
 or “PSXSHM” for a POSIX shared memory file;
 or “PUSG” for a */proc/usage* file;
 or “PW” for a */proc/watch* file;
 or “PXMP” for a */proc/xmap* file;
 or “REG” for a regular file;
 or “SMT” for a shared memory transport file;
 or “STSO” for a stream socket;
 or “UNNM” for an unnamed type file;
 or “XNAM” for an OpenServer Xenix special file of unknown type;
 or “XSEM” for an OpenServer Xenix semaphore file;
 or “XSD” for an OpenServer Xenix shared data file;
 or the four type number octets if the corresponding name isn’t known.

FILE-ADDR contains the kernel file structure address when **f** has been specified to **+f**;
 FCT contains the file reference count from the kernel file structure when **c** has been specified to **+f**;
 FILE-FLAG when **g** or **G** has been specified to **+f**, this field contains the contents of the `f_flag[s]` member of the kernel file structure and the kernel’s per-process open file flags (if available); ‘G’ causes them to be displayed in hexadecimal; ‘g’, as short-hand names; two lists may be displayed with entries separated by commas, the lists separated by a semi-colon (;); the first list may contain short-hand names for `f_flag[s]` values from the following table:

| | |
|------|---------------------------------|
| AIO | asynchronous I/O (e.g., FAIO) |
| AP | append |
| ASYN | asynchronous I/O (e.g., FASYNC) |
| BAS | block, test, and set in use |
| BKIU | block if in use |
| BL | use block offsets |
| BSK | block seek |
| CA | copy avoid |
| CIO | concurrent I/O |
| CLON | clone |
| CLRD | CL read |
| CR | create |
| DF | defer |
| DFI | defer IND |
| DFLU | data flush |
| DIR | direct |
| DLY | delay |
| DOCL | do clone |

| | |
|------|-------------------------------|
| DSYN | data-only integrity |
| DTY | must be a directory |
| EVO | event only |
| EX | open for exec |
| EXCL | exclusive open |
| FSYN | synchronous writes |
| GCDF | defer during unp_gc() (AIX) |
| GCMK | mark during unp_gc() (AIX) |
| GTTY | accessed via /dev/tty |
| HUP | HUP in progress |
| KERN | kernel |
| KIOC | kernel-issued ioctl |
| LCK | has lock |
| LG | large file |
| MBLK | stream message block |
| MK | mark |
| MNT | mount |
| MSYN | multiplex synchronization |
| NATM | don't update atime |
| NB | non-blocking I/O |
| NBDR | no BDRM check |
| NBIO | SYSV non-blocking I/O |
| NBF | n-buffering in effect |
| NC | no cache |
| ND | no delay |
| NDSY | no data synchronization |
| NET | network |
| NFLK | don't follow links |
| NMFS | NM file system |
| NOTO | disable background stop |
| NSH | no share |
| NTTY | no controlling TTY |
| OLRM | OLR mirror |
| PAIO | POSIX asynchronous I/O |
| PP | POSIX pipe |
| R | read |
| RC | file and record locking cache |
| REV | revoked |
| RSH | shared read |
| RSYN | read synchronization |
| RW | read and write access |
| SL | shared lock |
| SNAP | cooked snapshot |
| SOCK | socket |
| SQSH | Sequent shared set on open |
| SQSV | Sequent SVM set on open |
| SQR | Sequent set repair on open |
| SQS1 | Sequent full shared open |
| SQS2 | Sequent partial shared open |
| STPI | stop I/O |
| SWR | synchronous read |
| SYN | file integrity while writing |
| TCPM | avoid TCP collision |
| TR | truncate |

| | |
|------|------------------------------|
| W | write |
| WKUP | parallel I/O synchronization |
| WTG | parallel I/O synchronization |
| VH | vhangup pending |
| VTXT | virtual text |
| XL | exclusive lock |

this list of names was derived from F* #define's in dialect header files <fcntl.h>, <linux/fs.h>, <sys/fcntl.c>, <sys/fcntlcom.h>, and <sys/file.h>; see the lsof.h header file for a list showing the correspondence between the above short-hand names and the header file definitions;

the second list (after the semicolon) may contain short-hand names for kernel per-process open file flags from this table:

| | |
|------|------------------------------------|
| ALLC | allocated |
| BR | the file has been read |
| BHUP | activity stopped by SIGHUP |
| BW | the file has been written |
| CLSG | closing |
| CX | close-on-exec (see fcntl(F_SETFD)) |
| LCK | lock was applied |
| MP | memory-mapped |
| OPIP | open pending – in progress |
| RSVW | reserved wait |
| SHMT | UF_FSHMAT set (AIX) |
| USE | in use (multi-threaded) |

NODE-ID (or INODE-ADDR for some dialects) contains a unique identifier for the file node (usually the kernel vnode or inode address, but also occasionally a concatenation of device and node number) when **n** has been specified to **+f**;

DEVICE contains the device numbers, separated by commas, for a character special, block special, regular, directory or NFS file;

or “memory” for a memory file system node under Tru64 UNIX;

or the address of the private data area of a Solaris socket stream;

or a kernel reference address that identifies the file (The kernel reference address may be used for FIFO's, for example.);

or the base address or device name of a Linux AX.25 socket device.

Usually only the lower thirty two bits of Tru64 UNIX kernel addresses are displayed.

SIZE, SIZE/OFF, or OFFSET

is the size of the file or the file offset in bytes. A value is displayed in this column only if it is available. *Lsof* displays whatever value – size or offset – is appropriate for the type of the file and the version of *Lsof*.

On some UNIX dialects *Lsof* can't obtain accurate or consistent file offset information from its kernel data sources, sometimes just for particular kinds of files (e.g., socket files.) In other cases, files don't have true sizes – e.g., sockets, FIFOs, pipes – so *Lsof* displays for their sizes the content amounts it finds in their kernel buffer descriptors (e.g., socket buffer size counts or TCP/IP window sizes.) Consult the *Lsof* FAQ (The **FAQ** section gives its location.) for more information.

The file size is displayed in decimal; the offset is normally displayed in decimal with a leading “0t” if it contains 8 digits or less; in hexadecimal with a leading “0x” if it is longer than 8 digits. (Consult the **-o o** option description for information on when 8 might default to some other value.)

Thus the leading “0t” and “0x” identify an offset when the column may contain both a size and an offset (i.e., its title is SIZE/OFF).

If the **-o** option is specified, *lsOf* always displays the file offset (or nothing if no offset is available) and labels the column OFFSET. The offset always begins with “0t” or “0x” as described above.

The *lsOf* user can control the switch from “0t” to “0x” with the **-o o** option. Consult its description for more information.

If the **-s** option is specified, *lsOf* always displays the file size (or nothing if no size is available) and labels the column SIZE. The **-o** and **-s** options are mutually exclusive; they can’t both be specified.

For files that don’t have a fixed size – e.g., don’t reside on a disk device – *lsOf* will display appropriate information about the current size or position of the file if it is available in the kernel structures that define the file.

| | |
|-------|--|
| NLINK | contains the file link count when +L has been specified; |
| NODE | is the node number of a local file; or the inode number of an NFS file in the server host; or the Internet protocol type – e. g, “TCP”; or “STR” for a stream; or “CCITT” for an HP-UX x.25 socket; or the IRQ or inode number of a Linux AX.25 socket device. |
| NAME | is the name of the mount point and file system on which the file resides; or the name of a file specified in the <i>names</i> option (after any symbolic links have been resolved); or the name of a character special or block special device; or the local and remote Internet addresses of a network file; the local host name or IP number is followed by a colon (‘:’), the port, “->”, and the two-part remote address; IP addresses may be reported as numbers or names, depending on the + -M , -n , and -P options; colon-separated IPv6 numbers are enclosed in square brackets; IPv4 INADDR_ANY and IPv6 IN6_IS_ADDR_UNSPECIFIED addresses, and zero port numbers are represented by an asterisk (‘*’); a UDP destination address may be followed by the amount of time elapsed since the last packet was sent to the destination; TCP, UDP and UDPLITE remote addresses may be followed by TCP/TPI information in parentheses – state (e.g., “(ESTABLISHED)”, “(Unbound)”), queue sizes, and window sizes (not all dialects) – in a fashion similar to what <i>netstat</i> (1) reports; see the -T option description or the description of the TCP/TPI field in OUTPUT FOR OTHER PROGRAMS for more information on state, queue size, and window size; or the address or name of a UNIX domain socket, possibly including a stream clone device name, a file system object’s path name, local and foreign kernel addresses, socket pair information, and a bound vnode address; or the local and remote mount point names of an NFS file; or “STR”, followed by the stream name; or a stream character device name, followed by “->” and the stream name or a list of stream module names, separated by “->”; or “STR:” followed by the SCO OpenServer stream device and module names, separated by “->”; |

or system directory name, “--”, and as many components of the path name as *lsOf* can find in the kernel’s name cache for selected dialects (See the **KERNEL NAME CACHE** section for more information.);

or “PIPE->”, followed by a Solaris kernel pipe destination address;

or “COMMON:”, followed by the vnode device information structure’s device name, for a Solaris common vnode;

or the address family, followed by a slash (/), followed by fourteen comma-separated bytes of a non-Internet raw socket address;

or the HP-UX x.25 local address, followed by the virtual connection number (if any), followed by the remote address (if any);

or “(dead)” for disassociated Tru64 UNIX files – typically terminal files that have been flagged with the TIOCNOTTY ioctl and closed by daemons;

or “rd=<offset>” and “wr=<offset>” for the values of the read and write offsets of a FIFO;

or “clone *n*:/dev/event” for SCO OpenServer file clones of the /dev/event device, where *n* is the minor device number of the file;

or “(socketpair: *n*)” for a Solaris 2.6, 8, 9 or 10 UNIX domain socket, created by the *socketpair*(3N) network function;

or “no PCB” for socket files that do not have a protocol block associated with them, optionally followed by “, CANTSENDMORE” if sending on the socket has been disabled, or “, CANTRCVMORE” if receiving on the socket has been disabled (e.g., by the *shutdown*(2) function);

or the local and remote addresses of a Linux IPX socket file in the form <net>:[<node>:]<port>, followed in parentheses by the transmit and receive queue sizes, and the connection state;

or “dgram” or “stream” for the type UnixWare 7.1.1 and above in-kernel UNIX domain sockets, followed by a colon (:) and the local path name when available, followed by “->” and the remote path name or kernel socket address in hexadecimal when available;

or the association value, association index, endpoint value, local address, local port, remote address and remote port for Linux SCTP sockets;

or “protocol: ” followed by the Linux socket’s protocol attribute.

For dialects that support a “namefs” file system, allowing one file to be attached to another with *fattach*(3C), *lsOf* will add “(FA:<address1><direction><address2>)” to the NAME column. <address1> and <address2> are hexadecimal vnode addresses. <direction> will be “<-” if <address2> has been *fattach*’ed to this vnode whose address is <address1>; and “->” if <address1>, the vnode address of this vnode, has been *fattach*’ed to <address2>. <address1> may be omitted if it already appears in the DEVICE column.

lsOf may add two parenthetical notes to the NAME column for open Solaris 10 files: “(?)” if *lsOf* considers the path name of questionable accuracy; and “(deleted)” if the -X option has been specified and *lsOf* detects the open file’s path name has been deleted. Consult the *lsOf* FAQ (The **FAQ** section gives its location.) for more information on these NAME column additions.

LOCKS

lsOf can’t adequately report the wide variety of UNIX dialect file locks in a single character. What it reports in a single character is a compromise between the information it finds in the kernel and the limitations of the reporting format.

Moreover, when a process holds several byte level locks on a file, *lsOf* only reports the status of the first lock it encounters. If it is a byte level lock, then the lock character will be reported in lower case – i.e., ‘r’, ‘w’, or ‘x’ – rather than the upper case equivalent reported for a full file lock.

Generally *lsOf* can only report on locks held by local processes on local files. When a local process sets a lock on a remotely mounted (e.g., NFS) file, the remote server host usually records the lock state. One exception is Solaris – at some patch levels of 2.3, and in all versions above 2.4, the Solaris kernel records information on remote locks in local structures.

Lsof has trouble reporting locks for some UNIX dialects. Consult the **BUGS** section of this manual page or the *lsOf* FAQ (The **FAQ** section gives its location.) for more information.

OUTPUT FOR OTHER PROGRAMS

When the **-F** option is specified, *lsOf* produces output that is suitable for processing by another program – e.g, an *awk* or *Perl* script, or a C program.

Each unit of information is output in a field that is identified with a leading character and terminated by a NL (012) (or a NUL (000) if the 0 (zero) field identifier character is specified.) The data of the field follows immediately after the field identification character and extends to the field terminator.

It is possible to think of field output as process and file sets. A process set begins with a field whose identifier is ‘p’ (for process IDentifier (PID)). It extends to the beginning of the next PID field or the beginning of the first file set of the process, whichever comes first. Included in the process set are fields that identify the command, the process group IDentification (PGID) number, the task (thread) ID (TID), and the user ID (UID) number or login name.

A file set begins with a field whose identifier is ‘f’ (for file descriptor). It is followed by lines that describe the file’s access mode, lock state, type, device, size, offset, inode, protocol, name and stream module names. It extends to the beginning of the next file or process set, whichever comes first.

When the NUL (000) field terminator has been selected with the 0 (zero) field identifier character, *lsOf* ends each process and file set with a NL (012) character.

Lsof always produces one field, the PID (‘p’) field. All other fields may be declared optionally in the field identifier character list that follows the **-F** option. When a field selection character identifies an item *lsOf* does not normally list – e.g., PPID, selected with **-R** – specification of the field character – e.g., “**-FR**” – also selects the listing of the item.

It is entirely possible to select a set of fields that cannot easily be parsed – e.g., if the field descriptor field is not selected, it may be difficult to identify file sets. To help you avoid this difficulty, *lsOf* supports the **-F** option; it selects the output of all fields with NL terminators (the **-F0** option pair selects the output of all fields with NUL terminators). For compatibility reasons neither **-F** nor **-F0** select the raw device field.

These are the fields that *lsOf* will produce. The single character listed first is the field identifier.

| | |
|---|---|
| a | file access mode |
| c | process command name (all characters from proc or user structure) |
| C | file structure share count |
| d | file’s device character code |
| D | file’s major/minor device number (0x<hexadecimal>) |
| f | file descriptor (always selected) |
| F | file structure address (0x<hexadecimal>) |
| G | file flaGs (0x<hexadecimal>; names if +fg follows) |
| g | process group ID |
| i | file’s inode number |
| K | task ID |
| k | link count |
| l | file’s lock status |
| L | process login name |
| m | marker between repeated output |
| n | file name, comment, Internet address |
| N | node identifier (0x<hexadecimal>) |
| o | file’s offset (decimal) |

| | |
|-----|--|
| p | process ID (always selected) |
| P | protocol name |
| r | raw device number (0x<hexadecimal>) |
| R | parent process ID |
| s | file's size (decimal) |
| S | file's stream identification |
| t | file's type |
| T | TCP/TPI information, identified by prefixes (the '=' is part of the prefix): QR=<read queue size> QS=<send queue size> SO=<socket options and values> (not all dialects) SS=<socket states> (not all dialects) ST=<connection state> TF=<TCP flags and values> (not all dialects) WR=<window read size> (not all dialects) WW=<window write size> (not all dialects) (TCP/TPI information isn't reported for all supported UNIX dialects. The -h or -? help output for the -T option will show what TCP/TPI reporting can be requested.) |
| u | process user ID |
| z | Solaris 10 and higher zone name |
| Z | SELinux security context (inhibited when SELinux is disabled) |
| 0 | use NUL field terminator character in place of NL |
| 1-9 | dialect-specific field identifiers (The output of -F? identifies the information to be found in dialect-specific fields.) |

You can get on-line help information on these characters and their descriptions by specifying the **-F?** option pair. (Escape the '?' character as your shell requires.) Additional information on field content can be found in the **OUTPUT** section.

As an example, "**-F pcfm**" will select the process ID ('p'), command name ('c'), file descriptor ('f') and file name ('n') fields with an NL field terminator character; "**-F pcfm0**" selects the same output with a NUL (000) field terminator character.

Lsof doesn't produce all fields for every process or file set, only those that are available. Some fields are mutually exclusive: file device characters and file major/minor device numbers; file inode number and protocol name; file name and stream identification; file size and offset. One or the other member of these mutually exclusive sets will appear in field output, but not both.

Normally *lsof* ends each field with a NL (012) character. The 0 (zero) field identifier character may be specified to change the field terminator character to a NUL (000). A NUL terminator may be easier to process with *xargs* (1), for example, or with programs whose quoting mechanisms may not easily cope with the range of characters in the field output. When the NUL field terminator is in use, *lsof* ends each process and file set with a NL (012).

Three aids to producing programs that can process *lsof* field output are included in the *lsof* distribution. The first is a C header file, *lsof_fields.h*, that contains symbols for the field identification characters, indexes for storing them in a table, and explanation strings that may be compiled into programs. *Lsof* uses this header file.

The second aid is a set of sample scripts that process field output, written in *awk*, *Perl* 4, and *Perl* 5. They're located in the *scripts* subdirectory of the *lsof* distribution.

The third aid is the C library used for the *lsof* test suite. The test suite is written in C and uses field output to validate the correct operation of *lsof*. The library can be found in the *tests/LTlib.c* file of the *lsof* distribution. The library uses the first aid, the *lsof_fields.h* header file.

BLOCKS AND TIMEOUTS

Lsof can be blocked by some kernel functions that it uses – *lstat(2)*, *readlink(2)*, and *stat(2)*. These functions are stalled in the kernel, for example, when the hosts where mounted NFS file systems reside become inaccessible.

Lsof attempts to break these blocks with timers and child processes, but the techniques are not wholly reliable. When *Lsof* does manage to break a block, it will report the break with an error message. The messages may be suppressed with the **-t** and **-w** options.

The default timeout value may be displayed with the **-h** or **-?** option, and it may be changed with the **-S [t]** option. The minimum for *t* is two seconds, but you should avoid small values, since slow system responsiveness can cause short timeouts to expire unexpectedly and perhaps stop *Lsof* before it can produce any output.

When *Lsof* has to break a block during its access of mounted file system information, it normally continues, although with less information available to display about open files.

Lsof can also be directed to avoid the protection of timers and child processes when using the kernel functions that might block by specifying the **-O** option. While this will allow *Lsof* to start up with less overhead, it exposes *Lsof* completely to the kernel situations that might block it. Use this option cautiously.

AVOIDING KERNEL BLOCKS

You can use the **-b** option to tell *Lsof* to avoid using kernel functions that would block. Some cautions apply.

First, using this option usually requires that your system supply alternate device numbers in place of the device numbers that *Lsof* would normally obtain with the *lstat(2)* and *stat(2)* kernel functions. See the **ALTERNATE DEVICE NUMBERS** section for more information on alternate device numbers.

Second, you can't specify *names* for *Lsof* to locate unless they're file system names. This is because *Lsof* needs to know the device and inode numbers of files listed with *names* in the *Lsof* options, and the **-b** option prevents *Lsof* from obtaining them. Moreover, since *Lsof* only has device numbers for the file systems that have alternates, its ability to locate files on file systems depends completely on the availability and accuracy of the alternates. If no alternates are available, or if they're incorrect, *Lsof* won't be able to locate files on the named file systems.

Third, if the names of your file system directories that *Lsof* obtains from your system's mount table are symbolic links, *Lsof* won't be able to resolve the links. This is because the **-b** option causes *Lsof* to avoid the kernel *readlink(2)* function it uses to resolve symbolic links.

Finally, using the **-b** option causes *Lsof* to issue warning messages when it needs to use the kernel functions that the **-b** option directs it to avoid. You can suppress these messages by specifying the **-w** option, but if you do, you won't see the alternate device numbers reported in the warning messages.

ALTERNATE DEVICE NUMBERS

On some dialects, when *Lsof* has to break a block because it can't get information about a mounted file system via the *lstat(2)* and *stat(2)* kernel functions, or because you specified the **-b** option, *Lsof* can obtain some of the information it needs – the device number and possibly the file system type – from the system mount table. When that is possible, *Lsof* will report the device number it obtained. (You can suppress the report by specifying the **-w** option.)

You can assist this process if your mount table is supported with an */etc/mstab* or */etc/mnttab* file that contains an options field by adding a "dev=xxxx" field for mount points that do not have one in their options strings. Note: you must be able to edit the file – i.e., some mount tables like recent Solaris */etc/mnttab* or Linux */proc/mounts* are read-only and can't be modified.

You may also be able to supply device numbers using the **+m** and **+m m** options, provided they are supported by your dialect. Check the output of *Lsof*'s **-h** or **-?** options to see if the **+m** and **+m m** options are available.

The "xxxx" portion of the field is the hexadecimal value of the file system's device number. (Consult the *st_dev* field of the output of the *lstat(2)* and *stat(2)* functions for the appropriate values for your file

systems.) Here's an example from a Sun Solaris 2.6 */etc/mnttab* for a file system remotely mounted via NFS:

```
nfs ignore,noquota,dev=2a40001
```

There's an advantage to having "dev=xxxx" entries in your mount table file, especially for file systems that are mounted from remote NFS servers. When a remote server crashes and you want to identify its users by running *lsdf* on one of its clients, *lsdf* probably won't be able to get output from the *lstat(2)* and *stat(2)* functions for the file system. If it can obtain the file system's device number from the mount table, it will be able to display the files open on the crashed NFS server.

Some dialects that do not use an ASCII */etc/mntab* or */etc/mnttab* file for the mount table may still provide an alternative device number in their internal mount tables. This includes AIX, Apple Darwin, FreeBSD, NetBSD, OpenBSD, and Tru64 UNIX. *lsdf* knows how to obtain the alternative device number for these dialects and uses it when its attempt to *lstat(2)* or *stat(2)* the file system is blocked.

If you're not sure your dialect supplies alternate device numbers for file systems from its mount table, use this *lsdf* incantation to see if it reports any alternate device numbers:

```
lsdf -b
```

Look for standard error file warning messages that begin "assuming "dev=xxxx" from ...".

KERNEL NAME CACHE

lsdf is able to examine the kernel's name cache or use other kernel facilities (e.g., the ADVFS 4.x *tag_to_path()* function under Tru64 UNIX) on some dialects for most file system types, excluding AFS, and extract recently used path name components from it. (AFS file system path lookups don't use the kernel's name cache; some Solaris VxFS file system operations apparently don't use it, either.)

lsdf reports the complete paths it finds in the NAME column. If *lsdf* can't report all components in a path, it reports in the NAME column the file system name, followed by a space, two '-' characters, another space, and the name components it has located, separated by the '/' character.

When *lsdf* is run in repeat mode – i.e., with the *-r* option specified – the extent to which it can report path name components for the same file may vary from cycle to cycle. That's because other running processes can cause the kernel to remove entries from its name cache and replace them with others.

lsdf's use of the kernel name cache to identify the paths of files can lead it to report incorrect components under some circumstances. This can happen when the kernel name cache uses device and node number as a key (e.g., SCO OpenServer) and a key on a rapidly changing file system is reused. If the UNIX dialect's kernel doesn't purge the name cache entry for a file when it is unlinked, *lsdf* may find a reference to the wrong entry in the cache. The *lsdf* FAQ (The **FAQ** section gives its location.) has more information on this situation.

lsdf can report path name components for these dialects:

```
FreeBSD
HP-UX
Linux
NetBSD
NEXTSTEP
OpenBSD
OPENSTEP
SCO OpenServer
SCO|Caldera UnixWare
Solaris
Tru64 UNIX
```

lsdf can't report path name components for these dialects:

```
AIX
```

If you want to know why *lsdf* can't report path name components for some dialects, see the *lsdf* FAQ (The

FAQ section gives its location.)

DEVICE CACHE FILE

Examining all members of the */dev* (or */devices*) node tree with *stat(2)* functions can be time consuming. What's more, the information that *lsOf* needs – device number, inode number, and path – rarely changes.

Consequently, *lsOf* normally maintains an ASCII text file of cached */dev* (or */devices*) information (exception: the */proc*-based Linux *lsOf* where it's not needed.) The local system administrator who builds *lsOf* can control the way the device cache file path is formed, selecting from these options:

- Path from the **-D** option;
- Path from an environment variable;
- System-wide path;
- Personal path (the default);
- Personal path, modified by an environment variable.

Consult the output of the **-h**, **-D?**, or **-?** help options for the current state of device cache support. The help output lists the default read-mode device cache file path that is in effect for the current invocation of *lsOf*. The **-D?** option output lists the read-only and write device cache file paths, the names of any applicable environment variables, and the personal device cache path format.

lsOf can detect that the current device cache file has been accidentally or maliciously modified by integrity checks, including the computation and verification of a sixteen bit Cyclic Redundancy Check (CRC) sum on the file's contents. When *lsOf* senses something wrong with the file, it issues a warning and attempts to remove the current cache file and create a new copy, but only to a path that the process can legitimately write.

The path from which a *lsOf* process may attempt to read a device cache file may not be the same as the path to which it can legitimately write. Thus when *lsOf* senses that it needs to update the device cache file, it may choose a different path for writing it from the path from which it read an incorrect or outdated version.

If available, the **-Dr** option will inhibit the writing of a new device cache file. (It's always available when specified without a path name argument.)

When a new device is added to the system, the device cache file may need to be recreated. Since *lsOf* compares the mtime of the device cache file with the mtime and ctime of the */dev* (or */devices*) directory, it usually detects that a new device has been added; in that case *lsOf* issues a warning message and attempts to rebuild the device cache file.

Whenever *lsOf* writes a device cache file, it sets its ownership to the real UID of the executing process, and its permission modes to 0600, this restricting its reading and writing to the file's owner.

LSOF PERMISSIONS THAT AFFECT DEVICE CACHE FILE ACCESS

Two permissions of the *lsOf* executable affect its ability to access device cache files. The permissions are set by the local system administrator when *lsOf* is installed.

The first and rarer permission is *setuid-root*. It comes into effect when *lsOf* is executed; its effective UID is then root, while its real (i.e., that of the logged-on user) UID is not. The *lsOf* distribution recommends that versions for these dialects run *setuid-root*.

HP-UX 11.11 and 11.23
Linux

The second and more common permission is *setgid*. It comes into effect when the effective group IDentification number (GID) of the *lsOf* process is set to one that can access kernel memory devices – e.g., “*knmem*”, “*sys*”, or “*system*”.

An *lsOf* process that has *setgid* permission usually surrenders the permission after it has accessed the kernel memory devices. When it does that, *lsOf* can allow more liberal device cache path formations. The *lsOf* distribution recommends that versions for these dialects run *setgid* and be allowed to surrender *setgid* permission.

AIX 5.[12] and 5.3-ML1

Apple Darwin 7.x Power Macintosh systems
 FreeBSD 4.x, 4.1x, 5.x and [6789].x for x86-based systems
 FreeBSD 5.x and [6789].x for Alpha, AMD64 and Sparc64-based systems
 HP-UX 11.00
 NetBSD 1.[456], 2.x and 3.x for Alpha, x86, and SPARC-based systems
 NEXTSTEP 3.[13] for NEXTSTEP architectures
 OpenBSD 2.[89] and 3.[0–9] for x86-based systems
 OPENSTEP 4.x
 SCO OpenServer Release 5.0.6 for x86-based systems
 SCO|Caldera UnixWare 7.1.4 for x86-based systems
 Solaris 2.6, 8, 9 and 10
 Tru64 UNIX 5.1

(Note: *lsOf* for AIX 5L and above needs `setuid=`root permission if its `-X` option is used.)

Lsof for these dialects does not support a device cache, so the permissions given to the executable don't apply to the device cache file.

Linux

DEVICE CACHE FILE PATH FROM THE `-D` OPTION

The `-D` option provides limited means for specifying the device cache file path. Its `?` function will report the read-only and write device cache file paths that *lsOf* will use.

When the `-D b`, `r`, and `u` functions are available, you can use them to request that the cache file be built in a specific location (`b[path]`); read but not rebuilt (`r[path]`); or read and rebuilt (`u[path]`). The `b`, `r`, and `u` functions are restricted under some conditions. They are restricted when the *lsOf* process is `setuid=`root. The path specified with the `r` function is always read-only, even when it is available.

The `b`, `r`, and `u` functions are also restricted when the *lsOf* process runs `setgid` and *lsOf* doesn't surrender the `setgid` permission. (See the **LSOF PERMISSIONS THAT AFFECT DEVICE CACHE FILE ACCESS** section for a list of implementations that normally don't surrender their `setgid` permission.)

A further `-D` function, `i` (for ignore), is always available.

When available, the `b` function tells *lsOf* to read device information from the kernel with the `stat(2)` function and build a device cache file at the indicated path.

When available, the `r` function tells *lsOf* to read the device cache file, but not update it. When a path argument accompanies `-Dr`, it names the device cache file path. The `r` function is always available when it is specified without a path name argument. If *lsOf* is not running `setuid=`root and surrenders its `setgid` permission, a path name argument may accompany the `r` function.

When available, the `u` function tells *lsOf* to attempt to read and use the device cache file. If it can't read the file, or if it finds the contents of the file incorrect or outdated, it will read information from the kernel, and attempt to write an updated version of the device cache file, but only to a path it considers legitimate for the *lsOf* process effective and real UIDs.

DEVICE CACHE PATH FROM AN ENVIRONMENT VARIABLE

Lsof's second choice for the device cache file is the contents of the `LSOFDEVCACHE` environment variable. It avoids this choice if the *lsOf* process is `setuid=`root, or the real UID of the process is root.

A further restriction applies to a device cache file path taken from the `LSOFDEVCACHE` environment variable: *lsOf* will not write a device cache file to the path if the *lsOf* process doesn't surrender its `setgid` permission. (See the **LSOF PERMISSIONS THAT AFFECT DEVICE CACHE FILE ACCESS** section for information on implementations that don't surrender their `setgid` permission.)

The local system administrator can disable the use of the `LSOFDEVCACHE` environment variable or change its name when building *lsOf*. Consult the output of `-D?` for the environment variable's name.

SYSTEM-WIDE DEVICE CACHE PATH

The local system administrator may choose to have a system-wide device cache file when building *lsOf*. That file will generally be constructed by a special system administration procedure when the system is booted or when the contents of */dev* or */devices* changes. If defined, it is *lsOf*'s third device cache file path choice.

You can tell that a system-wide device cache file is in effect for your local installation by examining the *lsOf* help option output – i.e., the output from the **-h** or **-?** option.

lsOf will never write to the system-wide device cache file path by default. It must be explicitly named with a **-D** function in a root-owned procedure. Once the file has been written, the procedure must change its permission modes to 0644 (owner-read and owner-write, group-read, and other-read).

PERSONAL DEVICE CACHE PATH (DEFAULT)

The default device cache file path of the *lsOf* distribution is one recorded in the home directory of the real UID that executes *lsOf*. Added to the home directory is a second path component of the form *.lsOf_hostname*.

This is *lsOf*'s fourth device cache file path choice, and is usually the default. If a system-wide device cache file path was defined when *lsOf* was built, this fourth choice will be applied when *lsOf* can't find the system-wide device cache file. This is the **only** time *lsOf* uses two paths when reading the device cache file.

The *hostname* part of the second component is the base name of the executing host, as returned by *gethostname(2)*. The base name is defined to be the characters preceding the first '.' in the *gethostname(2)* output, or all the *gethostname(2)* output if it contains no '.'.

The device cache file belongs to the user ID and is readable and writable by the user ID alone – i.e., its modes are 0600. Each distinct real user ID on a given host that executes *lsOf* has a distinct device cache file. The *hostname* part of the path distinguishes device cache files in an NFS-mounted home directory into which device cache files are written from several different hosts.

The personal device cache file path formed by this method represents a device cache file that *lsOf* will attempt to read, and will attempt to write should it not exist or should its contents be incorrect or outdated.

The **-Dr** option without a path name argument will inhibit the writing of a new device cache file.

The **-D?** option will list the format specification for constructing the personal device cache file. The conversions used in the format specification are described in the *00DCACHE* file of the *lsOf* distribution.

MODIFIED PERSONAL DEVICE CACHE PATH

If this option is defined by the local system administrator when *lsOf* is built, the LSOFPERSDCPATH environment variable contents may be used to add a component of the personal device cache file path.

The LSOFPERSDCPATH variable contents are inserted in the path at the place marked by the local system administrator with the "%p" conversion in the HASPERSDC format specification of the dialect's *machine.h* header file. (It's placed right after the home directory in the default *lsOf* distribution.)

Thus, for example, if LSOFPERSDCPATH contains "LSOF", the home directory is "/Homes/abe", the host name is "lsOf.itap.purdue.edu", and the HASPERSDC format is the default ("%h/%p.lsOf_%L"), the modified personal device cache file path is:

/Homes/abe/LSOF/.lsOf_vic

The LSOFPERSDCPATH environment variable is ignored when the *lsOf* process is setuid-root or when the real UID of the process is root.

lsOf will not write to a modified personal device cache file path if the *lsOf* process doesn't surrender setgid permission. (See the **LSOF PERMISSIONS THAT AFFECT DEVICE CACHE FILE ACCESS** section for a list of implementations that normally don't surrender their setgid permission.)

If, for example, you want to create a sub-directory of personal device cache file paths by using the LSOFPERSDCPATH environment variable to name it, and *lsOf* doesn't surrender its setgid permission, you will have to allow *lsOf* to create device cache files at the standard personal path and move them to your subdirectory with shell commands.

The local system administrator may: disable this option when *lsyf* is built; change the name of the environment variable from `LSOFPERSDCPATH` to something else; change the `HASPERSDC` format to include the personal path component in another place; or exclude the personal path component entirely. Consult the output of the `-D?` option for the environment variable's name and the `HASPERSDC` format specification.

DIAGNOSTICS

Errors are identified with messages on the standard error file.

Lsof returns a one (1) if any error was detected, including the failure to locate command names, file names, Internet addresses or files, login names, NFS files, PIDs, PGIDs, or UIDs it was asked to list. If the `-V` option is specified, *lsyf* will indicate the search items it failed to list.

It returns a zero (0) if no errors were detected and if it was able to list some information about all the specified search arguments.

When *lsyf* cannot open access to `/dev` (or `/devices`) or one of its subdirectories, or get information on a file in them with `stat(2)`, it issues a warning message and continues. That *lsyf* will issue warning messages about inaccessible files in `/dev` (or `/devices`) is indicated in its help output – requested with the `-h` or `>B -?` options – with the message:

Inaccessible /dev warnings are enabled.

The warning message may be suppressed with the `-w` option. It may also have been suppressed by the system administrator when *lsyf* was compiled by the setting of the `WARNDEVACCESS` definition. In this case, the output from the help options will include the message:

Inaccessible /dev warnings are disabled.

Inaccessible device warning messages usually disappear after *lsyf* has created a working device cache file.

EXAMPLES

For a more extensive set of examples, documented more fully, see the `00QUICKSTART` file of the *lsyf* distribution.

To list all open files, use:

`lsyf`

To list all open Internet, x.25 (HP-UX), and UNIX domain files, use:

`lsyf -i -U`

To list all open IPv4 network files in use by the process whose PID is 1234, use:

`lsyf -i 4 -a -p 1234`

Presuming the UNIX dialect supports IPv6, to list only open IPv6 network files, use:

`lsyf -i 6`

To list all files using any protocol on ports 513, 514, or 515 of host `wonderland.cc.purdue.edu`, use:

`lsyf -i @wonderland.cc.purdue.edu:513-515`

To list all files using any protocol on any port of `mace.cc.purdue.edu` (`cc.purdue.edu` is the default domain), use:

`lsyf -i @mace`

To list all open files for login name “abe”, or user ID 1234, or process 456, or process 123, or process 789, use:

`lsyf -p 456,123,789 -u 1234,abe`

To list all open files on device `/dev/hd4`, use:

`lsyf /dev/hd4`

To find the process that has `/u/abe/foo` open, use:

```
lsof /u/abe/foo
```

To send a SIGHUP to the processes that have /u/abe/bar open, use:

```
kill -HUP `lsof -t /u/abe/bar`
```

To find any open file, including an open UNIX domain socket file, with the name /dev/log, use:

```
lsof /dev/log
```

To find processes with open files on the NFS file system named /nfs/mount/point whose server is inaccessible, and presuming your mount table supplies the device number for /nfs/mount/point, use:

```
lsof -b /nfs/mount/point
```

To do the preceding search with warning messages suppressed, use:

```
lsof -bw /nfs/mount/point
```

To ignore the device cache file, use:

```
lsof -Di
```

To obtain PID and command name field output for each process, file descriptor, file device number, and file inode number for each file of each process, use:

```
lsof -FpcfDi
```

To list the files at descriptors 1 and 3 of every process running the *lsof* command for login ID “abe” every 10 seconds, use:

```
lsof -c lsof -a -d 1 -d 3 -u abe -r10
```

To list the current working directory of processes running a command that is exactly four characters long and has an ‘o’ or ‘O’ in character three, use this regular expression form of the *-c c* option:

```
lsof -c /^..o.$/i -a -d cwd
```

To find an IP version 4 socket file by its associated numeric dot-form address, use:

```
lsof -i@128.210.15.17
```

To find an IP version 6 socket file (when the UNIX dialect supports IPv6) by its associated numeric colon-form address, use:

```
lsof -i@[0:1:2:3:4:5:6:7]
```

To find an IP version 6 socket file (when the UNIX dialect supports IPv6) by an associated numeric colon-form address that has a run of zeroes in it – e.g., the loop-back address – use:

```
lsof -i@[::1]
```

To obtain a repeat mode marker line that contains the current time, use:

```
lsof -rm====%T====
```

To add spaces to the previous marker line, use:

```
lsof -r "m==== %T===="
```

BUGS

Since *lsof* reads kernel memory in its search for open files, rapid changes in kernel memory may produce unpredictable results.

When a file has multiple record locks, the lock status character (following the file descriptor) is derived from a test of the first lock structure, not from any combination of the individual record locks that might be described by multiple lock structures.

Lsof can’t search for files with restrictive access permissions by *name* unless it is installed with root set-UID permission. Otherwise it is limited to searching for files to which its user or its set-GID group (if any) has access permission.

The display of the destination address of a raw socket (e.g., for *ping*) depends on the UNIX operating system. Some dialects store the destination address in the raw socket's protocol control block, some do not.

Lsof can't always represent Solaris device numbers in the same way that *ls*(1) does. For example, the major and minor device numbers that the *lstat*(2) and *stat*(2) functions report for the directory on which CD-ROM files are mounted (typically */cdrom*) are not the same as the ones that it reports for the device on which CD-ROM files are mounted (typically */dev/sr0*). (*Lsof* reports the directory numbers.)

The support for */proc* file systems is available only for BSD and Tru64 UNIX dialects, Linux, and dialects derived from SYSV R4 – e.g., FreeBSD, NetBSD, OpenBSD, Solaris, UnixWare.

Some */proc* file items – device number, inode number, and file size – are unavailable in some dialects. Searching for files in a */proc* file system may require that the full path name be specified.

No text (**txt**) file descriptors are displayed for Linux processes. All entries for files other than the current working directory, the root directory, and numerical file descriptors are labeled **mem** descriptors.

Lsof can't search for Tru64 UNIX named pipes by name, because their kernel implementation of *lstat*(2) returns an improper device number for a named pipe.

Lsof can't report fully or correctly on HP-UX 9.01, 10.20, and 11.00 locks because of insufficient access to kernel data or errors in the kernel data. See the *lsof* FAQ (The **FAQ** section gives its location.) for details.

The AIX SMT file type is a fabrication. It's made up for file structures whose type (15) isn't defined in the AIX */usr/include/sys/file.h* header file. One way to create such file structures is to run X clients with the *DISPLAY* variable set to ":0.0".

The **+|-f[*cfgGn*]** option is not supported under */proc*-based Linux *lsof*, because it doesn't read kernel structures from kernel memory.

ENVIRONMENT

Lsof may access these environment variables.

| | |
|----------------|--|
| LANG | defines a language locale. See <i>setlocale</i> (3) for the names of other variables that can be used in place of LANG – e.g., LC_ALL, LC_TYPE, etc. |
| LSOFDEVCACHE | defines the path to a device cache file. See the DEVICE CACHE PATH FROM AN ENVIRONMENT VARIABLE section for more information. |
| LSOFPERSDCPATH | defines the middle component of a modified personal device cache file path. See the MODIFIED PERSONAL DEVICE CACHE PATH section for more information. |

FAQ

Frequently-asked questions and their answers (an FAQ) are available in the *00FAQ* file of the *lsof* distribution.

That file is also available via anonymous ftp from *lsof.itap.purdue.edu* at *pub/tools/unix/lsof/FAQ*. The URL is:

<ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/FAQ>

FILES

| | |
|-----------------------|--|
| <i>/dev/kmem</i> | kernel virtual memory device |
| <i>/dev/mem</i> | physical memory device |
| <i>/dev/swap</i> | system paging device |
| <i>.lsof_hostname</i> | <i>lsof</i> 's device cache file (The suffix, <i>hostname</i> , is the first component of the host's name returned by <i>gethostname</i> (2).) |

AUTHORS

Lsof was written by Victor A. Abell <abe@purdue.edu> of Purdue University. Many others have contributed to *lsof*. They're listed in the *00CREDITS* file of the *lsof* distribution.

DISTRIBUTION

The latest distribution of *lsof* is available via anonymous ftp from the host *lsof.itap.purdue.edu*. You'll find the *lsof* distribution in the *pub/tools/unix/lsof* directory.

You can also use this URL:

`ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof`

Lsof is also mirrored elsewhere. When you access *lsof.itap.purdue.edu* and change to its *pub/tools/unix/lsof* directory, you'll be given a list of some mirror sites. The *pub/tools/unix/lsof* directory also contains a more complete list in its *mirrors* file. Use mirrors with caution – not all mirrors always have the latest *lsof* revision.

Some pre-compiled *Lsof* executables are available on *lsof.itap.purdue.edu*, but their use is discouraged – it's better that you build your own from the sources. If you feel you must use a pre-compiled executable, please read the cautions that appear in the README files of the *pub/tools/unix/lsof/binaries* subdirectories and in the 00* files of the distribution.

More information on the *lsof* distribution can be found in its *README.lsof_<version>* file. If you intend to get the *lsof* distribution and build it, please read *README.lsof_<version>* and the other 00* files of the distribution before sending questions to the author.

SEE ALSO

Not all the following manual pages may exist in every UNIX dialect to which *lsof* has been ported.

`access(2)`, `awk(1)`, `crash(1)`, `fattach(3C)`, `ff(1)`, `fstat(8)`, `fuser(1)`, `gethostname(2)`, `isprint(3)`, `kill(1)`, `local-time(3)`, `lstat(2)`, `modload(8)`, `mount(8)`, `netstat(1)`, `ofiles(8L)`, `perl(1)`, `ps(1)`, `readlink(2)`, `setlocale(3)`, `stat(2)`, `strftime(3)`, `time(2)`, `uname(1)`.

NAME

lspm – list names and descriptions of Perl modules in a directory

SYNOPSIS

```
lspm -h
```

```
lspm [-p] [-a] [-c [num]] [-l len] [dir [dir dir ...]]
```

DESCRIPTION

Lists all or a subset of installed Perl modules, with version numbers and descriptions.

It will look in Perl's default search path for modules, @INC, if you don't explicitly list any directories to walk. Note that this default search excludes the current directory.

OPTIONS

-h, --help

See a synopsis.

--man

Browse the manpage.

-p, --show-path

Include path of found modules in output.

-a, --align

Vertically align descriptions.

-c, --align-local, --align-cont

Align descriptions in blocks where the module names don't differ too much in length, to avoid pushing all descriptions way over to the right just because a few names are long. The output looks more ragged than with full alignment, but is still lined up locally and only requires the eye to cross small gaps between columns, so is usually more readable.

You can pass an optional positive integer argument to specify the length threshold; the default is 7.

-l, --max-length, --limit

Cut off descriptions at specified length.

SEE ALSO

edrcintro(1), **pmdesc(1)**, <http://www.cpan.org/modules/by-authors/id/TOMC/scripts/pmdesc.gz>

BUGS

I need something to write here.

COPYRIGHT AND LICENCE

Written by Aristotle Pagaltzis, (c)2006.

This module is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

NAME

lua – Lua interpreter

SYNOPSIS

lua [*options*] [*script* [*args*]]

DESCRIPTION

lua is the standalone Lua interpreter. It loads and executes Lua programs, either in textual source form or in precompiled binary form. (Precompiled binaries are output by **luac**, the Lua compiler.) **lua** can be used as a batch interpreter and also interactively.

After handling the *options*, the Lua program in file *script* is loaded and executed. The *args* are available to *script* as strings in a global table named **arg** and also as arguments to its main function. When called without arguments, **lua** behaves as **lua -v -i** if the standard input is a terminal, and as **lua -** otherwise.

In interactive mode, **lua** prompts the user, reads lines from the standard input, and executes them as they are read. If the line contains an expression, then the line is evaluated and the result is printed. If a line does not contain a complete statement, then a secondary prompt is displayed and lines are read until a complete statement is formed or a syntax error is found.

Before handling command line options and scripts, **lua** checks the contents of the environment variables **LUA_INIT_5_4** and **LUA_INIT**, in that order. If the contents are of the form '@*filename*', then *filename* is executed. Otherwise, the contents are assumed to be a Lua statement and is executed. When **LUA_INIT_5_4** is defined, **LUA_INIT** is ignored.

OPTIONS

- e** *stat* execute statement *stat*.
- i** enter interactive mode after executing *script*.
- l** *name* require library *name* into global *name*.
- v** show version information.
- E** ignore environment variables.
- W** turn warnings on.
- stop handling options.
- stop handling options and execute the standard input as a file.

ENVIRONMENT VARIABLES

The following environment variables affect the execution of **lua**. When defined, the version-specific variants take priority and the version-neutral variants are ignored.

LUA_INIT, LUA_INIT_5_4

Code to be executed before command line options and scripts.

LUA_PATH, LUA_PATH_5_4

Initial value of package.cpath, the path used by require to search for Lua loaders.

LUA_CPATH, LUA_CPATH_5_4

Initial value of package.cpath, the path used by require to search for C loaders.

EXIT STATUS

If a script calls `os.exit`, then **lua** exits with the given exit status. Otherwise, **lua** exits with `EXIT_SUCCESS` (0 on POSIX systems) if there were no errors and with `EXIT_FAILURE` (1 on POSIX systems) if there were errors. Errors raised in interactive mode do not cause exits.

DIAGNOSTICS

Error messages should be self explanatory.

SEE ALSO

edrcintro(1), luac(1), luaenv(3), luaversion(3), lua_wrapper(1)

The documentation at lua.org, especially section 7 of the reference manual.

AUTHORS

R. Ierusalimschy, L. H. de Figueiredo, W. Celes

NAME

luac – Lua compiler

SYNOPSIS

luac [*options*] [*filenames*]

DESCRIPTION

luac is the Lua compiler. It translates programs written in the Lua programming language into binary files containing precompiled chunks that can be later loaded and executed.

The main advantages of precompiling chunks are: faster loading, protecting source code from accidental user changes, and off-line syntax checking. Precompiling does not imply faster execution because in Lua chunks are always compiled into bytecodes before being executed. **luac** simply allows those bytecodes to be saved in a file for later execution. Precompiled chunks are not necessarily smaller than the corresponding source. The main goal in precompiling is faster loading.

In the command line, you can mix text files containing Lua source and binary files containing precompiled chunks. **luac** produces a single output file containing the combined bytecodes for all files given. Executing the combined file is equivalent to executing the given files. By default, the output file is named **luac.out**, but you can change this with the **-o** option.

Precompiled chunks are *not* portable across different architectures. Moreover, the internal format of precompiled chunks is likely to change when a new version of Lua is released. Make sure you save the source files of all Lua programs that you precompile.

OPTIONS

- l** produce a listing of the compiled bytecode for Lua's virtual machine. Listing bytecodes is useful to learn about Lua's virtual machine. If no files are given, then **luac** loads **luac.out** and lists its contents. Use **-l -l** for a full listing.
- o file** output to *file*, instead of the default **luac.out**. (You can use '-' for standard output, but not on platforms that open standard output in text mode.) The output file may be one of the given files because all files are loaded before the output file is written. Be careful not to overwrite precious files.
- p** load files but do not generate any output file. Used mainly for syntax checking and for testing precompiled chunks: corrupted files will probably generate errors when loaded. If no files are given, then **luac** loads **luac.out** and tests its contents. No messages are displayed if the file loads without errors.
- s** strip debug information before writing the output file. This saves some space in very large chunks, but if errors occur when running a stripped chunk, then the error messages may not contain the full information they usually do. In particular, line numbers and names of local variables are lost.
- v** show version information.
- stop handling options.
- stop handling options and process standard input.

SEE ALSO

edrcintro(1), **lua(1)**, **luaenv(3)**, **luaversion(3)**, **lua_wrapper(1)**
The documentation at lua.org.

DIAGNOSTICS

Error messages should be self explanatory.

AUTHORS

R. Ierusalimschy, L. H. de Figueiredo, W. Celes

NAME

luaenv – print environment needed to start Lua scripts

SYNOPSIS

edrc/lib/luaenv [**-n**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print the environment used by **luaenv** to access the Lua packages/modules bundled with WA2L/edrc.

To set the environment prior to the execution of **lua**, invoke:

```
eval `luaenv`
```

This command is *only* to be used to prepare the environment to execute Lua scripts when calling the **lua** interpreter not thru **edrc/bin/lua**.

When writing Lua scripts for WA2L/edrc, do *not* use **luaenv**, start the script thru the **.lua_wrapper**. See **lua_wrapper(1)** for information about integrating a Lua script into WA2L/edrc.

When writing recovery scripts in Lua, use the:

```
#!/usr/bin/env lua
```

setting in the script magic key and the environment is set correctly.

OPTIONS

-n no **export** *VARIABLE* ... output.

ENVIRONMENT**\$LUA_PATH**

this variable holds Lua package/module locations additional to the standard applying to an installation. This variable will be expanded with the bundled Lua module locations (**lib/lua/lum/<OSID>/<major>.<minor>/share/lua/<major>.<minor>/?.lua;;**) when executing the **luaenv** command.

\$LUA_CPATH

path of the Lua package/module libs for the related operating system id and Lua version (**lib/lua/lum/<OSID>/<major>.<minor>/lib/lua/<major>.<minor>/[<GLIBC>/]?.so;;**) bundled with WA2L/edrc.

The GLIBC version *<GLIBC>* can be resolved using the **glibc.version(3)**, the **ldd --version** or the **getconf GNU_LIBC_VERSION** command.

\$PATH command search path. This variable will be expanded with the path where the **lua** interpreter was found on the system when executing the **luaenv** command. See also **lua_wrapper(1)** for more information.

EXIT STATUS

0 always

FILES

The *<major>* is the major version number (e.g. 5) and *<minor>* is the minor Lua version (e.g. 4).

etc/lua_wrapper.cfg

configuration file for **.lua_wrapper** and **luaenv**.

lib/lua/lum/<OSID>/<major>.<minor>/

location of all bundled Lua packages/modules.

lib/lua/lum/<OSID>/<major>.<minor>/share/lua/<major>.<minor>/

Pure Lua modules (**.lua** files).

The directory is pre-pended to **\$LUA_PATH**.

lib/lua/lum/<OSID>/<major>.<minor>/lib/lua/<major>.<minor>/[<GLIBC>/]

Compiled Lua modules (**.so** files).

The directory is pre-pended to the **\$LUA_CPATH**.

The logic of the libraries is identical to the **os_wrapper(1)** to support operating systems having different versions of the **glibc** library.

The GLIBC version *<GLIBC>* can be resolved using the **glibc.version(3)**, the **ldd --version** or the **getconf GNU_LIBC_VERSION** command.

lib/lua/lum/<OSID>/<major>.<minor>/bin/

Scripts provided by some Lua modules.

EXAMPLES

1) Start a Lua script from a recovery script

To start a Lua script that uses a bundled Lua module from a recovery script, just call the **lua** interpreter.

Recovery script:

```
#!/bin/ksh
#
# 1:ascript - A Recovery Script calling a Lua script
#
# [00] 08.02.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x
:
:
lua ./luascript
:
```

Lua script:

```
--
-- luascript - Lua script using bundled lcomplex module
--
-- [00] 18.03.2009 CWa      Initial Version
--
:
:
local complex = require "complex"
:
```

2) Write a recovery script in Lua

Write the recovery script in Lua.

Lua recovery script:

```
#!/usr/bin/env lua
--
-- 1:ascript - A Recovery Script in Lua
--
-- [00] 08.02.2009 CWa      Initial Version
--
:
:
local complex = require "complex"
:
```

Note the use of `'#!/usr/bin/env lua'` in the magic key to start the Lua script. This ensures that the **lua** interpreter is found independent of the installation base directory of WA2L/edrc.

3) Write a Lua script outside of WA2L/edrc and profit from bundled Lua modules

The startup method uses the **lua** interpreter start thru `~edrc/bin/lua` that automatically enables to use all Lua packages/modules bundled with WA2L/edrc.

This method is similar to the use of the **.lua_wrapper**, but the Lua script can be placed somewhere of your liking outside of WA2L/edrc.

Set the permissions of the script to executable:

```
chmod +x luascript
```

Lua script:

```
#!/bin/ksh
#
# luascript - Lua script using bundled modules
#
# [00] 18.03.2009 CWa Initial Version
#

sed '1,/^exit \$\$/d' $0 | ~edrc/bin/lua - "$0" "$@"
exit $?

-- Lua
--
:
:
local complex = require "complex"
:
```

4) Write a Lua script outside of WA2L/edrc and profit from bundled Lua modules

The startup method method below allows to start the **lua** interpreter from a specific installation directory on the system and also enables to use all Lua packages/modules bundled with WA2L/edrc using the **luaenv** command to dynamically initialize the environment.

This method is similar to the use of the **.lua_wrapper**, but the Lua script can be placed somewhere of your liking outside of WA2L/edrc.

Set the permissions of the script to executable:

```
chmod +x luascript
```

Lua script:

```
#!/bin/ksh
#
# luascript - Lua script using bundled modules
#
# [00] 18.03.2009 CWa Initial Version
#

eval `~edrc/lib/luaenv`

sed '1,/^exit \$\$/d' $0 | /usr/bin/lua - "$0" "$@"
exit $?
```

```
-- Lua
--
:
:
local complex = require "complex"
:
```

5) Test the availability of a specific Lua package/module

Try to access the Lua module:

```
[ / ]
[ root@acme007 ][*edrc*/bash]: ~edrc/bin/lua -e 'require "complex"'
```

If an error message like

```
/opt/edrc/bin/Linux-64/2.27/lua: (command line):1: module 'complex' not
no field package.preload['complex']
no file '/opt/edrc/lib/lua/lum/Linux-64/5.4/share/lua/5.4/complex.l
no file '/usr/local/share/lua/5.4/complex.lua'
no file '/usr/local/share/lua/5.4/complex/init.lua'
no file '/usr/local/lib/lua/5.4/complex.lua'
no file '/usr/local/lib/lua/5.4/complex/init.lua'
no file './complex.lua'
no file './complex/init.lua'
no file '/opt/edrc/lib/lua/lum/Linux-64/5.4/lib/complex.so'
no file '/usr/local/lib/lua/5.4/complex.so'
no file '/usr/local/lib/lua/5.4/loadall.so'
no file './complex.so'
stack traceback:
[C]: in function 'require'
(command line):1: in main chunk
[C]: in ?
```

appears, the package/module cannot be found, if the package/module can be found there is no output.

SEE ALSO

edrcintro(1), **glibc.version(3)**, **ldd(1)**, **lua(1)**, **luarocks(1)**, **luaversion(3)**, **lua_wrapper(1)**, **osid(3)**, **pkgdir(1m)**, <https://luarocks.org/>, <https://github.com/luarocks/luarocks/blob/main/docs/index.md>

NOTES

because the **lua** executable is bundled with WA2L/edrc the use of **luaenv** is not so extensively needed as **perlenv(3)** for Perl scripts or **pythonenv(3)** for Python scripts.

BUGS

-

AUTHOR

luaenv was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

luarocks - module deployment system for Lua

SYNOPSIS

luarocks [*options*] *command*

DESCRIPTION

LuaRocks is a module deployment system for Lua

Variables from the "variables" table of the configuration file can be overridden with VAR=VALUE assignments.

Options:

--from=<server>

Fetch rocks/rockspecs from this server (takes priority over config file)

--only-from=<server>

Fetch rocks/rockspecs from this server only (overrides any entries in the config file)

--to=<tree>

Which tree to operate on.

Commands:

build Build/compile a rock.

download

Download a specific rock file from a rocks server.

help Help on commands.

install Install a rock.

list Lists currently installed rocks.

make Compile package in current directory using a rockspec.

pack Create a rock, packing sources or binaries.

remove

Uninstall a rock.

search Query the LuaRocks servers.

unpack

Unpack the contents of a rock.

config Query configuration

SEE ALSO

[edrcintro\(1\)](#), [luaenv\(3\)](#), [luarocks.cfg\(4\)](#), [luaversion\(3\)](#), [lua_wrapper\(1\)](#), <https://luarocks.org/>, <https://github.com/luarocks/luarocks/blob/main/docs/index.md>

FILES

etc/luarocks.cfg

AUTHOR

Enrico Tassi <gareuselesinge@debian.org>

NAME

luarocks.cfg – configuration file for luarocks

SYNOPSIS

edrc/etc/luarocks.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **luarocks** command.

FILEFORMAT

In general the **luarocks.cfg** file is a **lua** file. However, not all language features and commands are available.

For a detail description of the configuration file syntax, see: <https://github.com/luarocks/luarocks/blob/main/docs/index.md>

The description given in this manual page reflects the settings needed to integrate Lua and **luarocks** properly into WA2L/edrc.

OPTIONS**ENVIRONMENT VARIABLES**

The following environment variables holding WA2L/edrc settings are available within the configuration file.

```
local APPROOT    = os_getenv("APPROOT")
local OSID       = os_getenv("OSID")
local LIBC       = os_getenv("LIBC")
local LUAVERSION = os_getenv("LUAVERSION")
```

Whereas **APPROOT** contains the output of **approot**, **OSID** the output of **osid**, **LIBC** the output of **glibc.version** and **LUAVERSION** contains the output of **luaversion**.

rocks_trees

Base directory of the Lua modules distributed with WA2L/edrc and handled by the **luarocks** command within WA2L/edrc.

```
rocks_trees = {
  { name = "system", root = APPROOT .. "/lib/lua/lum/" .. OSID .. "/"
```

```
}
```

lib_modules_path

Sub-directory in the **rocks_tree** with the name **system**. This directory holds compiled Lua package libraries (**.so** files).

```
lib_modules_path = "lib/lua/" .. LUAVERSION .. "/" .. LIBC
```

After installing a package/module that provides compiled libraries into the **lib_modules_path** directory **edrc/lib/lua/lum/<OSID>/<major>.<minor>/lib/lua/<major>.<minor>/<GLIBC>/** ensure to manually create the related symbolic links in **edrc/lib/lua/lum/<OSID>/<major>.<minor>/lib/lua/<major>.<minor>/** and **edrc/lib/lua/lum/<OSID>/<major>.<minor>/lib/lua/<major>.<minor>/<GLIBC>/** to provide maximal portability.

variables The **variables** section contain settings that point to several Lua components needed to build and install a Lua package.

```
variables = {
    LUA_DIR      = APPROOT .. "/bin/" .. OSID;
    LUA_INCDIR   = APPROOT .. "/lib/" .. OSID .. "/includes";
    LUA_BINDIR   = APPROOT .. "/bin/" .. OSID .. "/" .. LIBC;
    LUA_LIBDIR   = APPROOT .. "/lib/" .. OSID .. "/" .. LIBC .. "libs";
    LUA_VERSION  = LUAVERSION;
    LUA          = APPROOT .. "/bin/" .. OSID .. "/" .. LIBC .. "lua";
}
```

SEE ALSO

edrcintro(1), **approot(3)**, **glibc.version(3)**, **luaenv(3)**, **luarocks(1)**, **luaversion(3)**, **lua_wrapper(1)**, **osid(3)**, <https://luarocks.org/>, <https://github.com/luarocks/luarocks/blob/main/docs/index.md>

NOTES

-

BUGS

-

AUTHOR

luarocks.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

luaversion – print version (major.minor) of Lua interpreter

SYNOPSIS

edrc/lib/luaversion

AVAILABILITY

WA2L/edrc

DESCRIPTION

print Lua version of **lua** interpreter found on the system.

See also **lua_wrapper**(1) for additional information.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **appreversion**(3), **luaenv**(3), **lua_wrapper**(1), **revision**(3), **scriptrevision**(3)

NOTES

-

BUGS

-

AUTHOR

luaversion was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lua_wrapper – wrap Lua scripts to ensure lua startup

SYNOPSIS

edrc/bin/luaprogram -> .lua_wrapper

edrc/lib/luaprogram -> .lua_wrapper

AVAILABILITY

WA2L/edrc

DESCRIPTION

Wrap Lua scripts/programs to avoid using `#!/usr/bin/lua` or similar path names in the magic key. This enhances the portability of WA2L/edrc commands written in Lua to systems where **lua** is not installed in the default install locations or to use **lua** distributed with WA2L/edrc.

lua has to be installed in **edrc/bin/**, **/usr/bin/**, **/bin/**, **/sbin/**, **/usr/local/bin/** or **/usr/contrib/bin/**.

Further **lua** locations can be configured in the optional config file **lua_wrapper.cfg**.

The Lua packages/modules bundled with WA2L/edrc that are installed in **lib/lua/lum/<OSID>/<major>.<minor>/** are included in the Lua path (**LUA_PATH**, **LUA_CPATH**).

To start a new command thru the **.lua_wrapper**, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ][*edrc*/bash]: ln -s .lua_wrapper new_cmd
```

- 2.) install the Lua script/program in the **edrc/lib/lua/** directory:

```
[ /opt/edrc/lib/lua ]
[ root@acme001 ][*edrc*/bash]: vi new_cmd

[ /opt/edrc/lib/lua ]
[ root@acme001 ][*edrc*/bash]: chmod 644 new_cmd
```

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- 101** shell (**lua**) not found. If **lua** is installed on the system and you get this error, add the **lua** location in the configuration file **lua_wrapper.cfg**.
- 102** the Lua script to be started that should be located in **lib/lua/** does not exist.
- 103** user calling the command has no permission to access/execute the called *lua* program.
- 107** the **.lua_wrapper** was called directly.

FILES

For a detailed description of the **FILES**, see: **luaenv(3)**.

etc/lua_wrapper.cfg

configuration file for the **.lua_wrapper** command.

lib/lua/

location of the wrapped Lua scripts/programs. This files should have the file permissions *644* to show, that those scripts should not be started directly.

lib/lua/lum/<OSID>/<major>.<minor>/

location of the Lua packages/modules bundled with WA2L/edrc.

EXAMPLES

-

SEE ALSO

edrcintro(1), **binprobe(1m)**, **cmdlist(1m)**, **daemon_wrapper(1)**, **java_wrapper(1)**, **ksh_wrapper(1)**, **ld(1)**, **ln(1)**, **lua(1)**, **luac(1)**, **luaenv(3)**, **luaversion(3)**, **lua_wrapper.cfg(4)**, **osid(3)**, **os_wrapper(1)**, **python_wrapper(1)**, **perl_wrapper(1)**

NOTES

-

BUGS

-

AUTHOR

lua_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lua_wrapper.cfg – configuration file for `.lua_wrapper`

SYNOPSIS

`edrc/etc/lua_wrapper.cfg`

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the `.lua_wrapper` command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SEARCH_PATH**

Colon separated search path where to search for the **lua** interpreter.

Example: `SEARCH_PATH="/usr/bin:/bin:/sbin:/usr/local/bin:/usr/contrib/bin"`

Default: `SEARCH_PATH="`approot`/bin:/usr/bin:/bin:/sbin:/usr/local/bin:/usr/contrib/bin"`

SEE ALSO

`edrcintro(1)`, `lua_wrapper(1)`

NOTES

-

BUGS

-

AUTHOR

lua_wrapper.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

lynx – a general purpose distributed information browser for the World Wide Web

SYNOPSIS

lynx [*options*] [*optional paths or URLs*]

lynx [*options*] [*path or URL*] **-get_data**
data

lynx [*options*] [*path or URL*] **-post_data**
data

Use “lynx -help” to display a complete list of current options.

DESCRIPTION

Lynx is a fully-featured World Wide Web (WWW) client for users running cursor-addressable, character-cell display devices (e.g., vt100 terminals, vt100 emulators running on Windows 95/NT/XP/7/8 or any POSIX platform, or any other “curses-oriented” display). It will display hypertext markup language (HTML) documents containing links to files residing on the local system, as well as files residing on remote systems running Gopher, HTTP, FTP, WAIS, and NNTP servers. Current versions of *Lynx* run on Unix, VMS, Windows 95/NT/XP/7/8, DOS DJGPP and OS/2.

Lynx can be used to access information on the World Wide Web, or to build information systems intended primarily for local access. For example, *Lynx* has been used to build several Campus Wide Information Systems (CWIS). In addition, *Lynx* can be used to build systems isolated within a single LAN.

OPTIONS

At start up, *Lynx* will load any local file or remote URL specified at the command line. For help with URLs, press “?” or “H” while running *Lynx*. Then follow the link titled, “Help on URLs.”

If more than one local file or remote URL is listed on the command line, *Lynx* will open only the last interactively. All of the names (local files and remote URLs) are added to the **G)oto** history.

Lynx uses only long option names. Option names can begin with double dash “--” as well, underscores and dashes can be intermixed in option names (in the reference below, options are shown with one dash “-” before them, and with underscores “_”).

Lynx provides many command-line options. Some options require a value (string, number or keyword). These are noted in the reference below. The other options set boolean values in the program. There are three types of boolean options: set, unset and toggle. If no option value is given, these have the obvious meaning: set (to true), unset (to false), or toggle (between true/false). For any of these, an explicit value can be given in different forms to allow for operating system constraints, e.g.,

```
-center:off
-center=off
-center-
```

Lynx recognizes “1”, “+”, “on” and “true” for true values, and “0”, “-”, “off” and “false” for false values. Other option-values are ignored.

The default boolean, number and string option values that are compiled into *Lynx* are displayed in the help-message provided by **lynx -help**. Some of those may differ according to how *Lynx* was built; see the help message itself for these values. The **-help** option is processed in the third pass of options-processing, so any option which sets a value, as well as runtime configuration values are reflected in the help-message.

- If the argument is only “-”, then *Lynx* expects to receive the arguments from the standard input. This is to allow for the potentially very long command line that can be associated with the **-get_data** or **-post_data** arguments (see below). It can also be used to avoid having sensitive information in the invoking command line (which would be visible to other processes on most systems), especially when the **-auth** or **-pauth** options are used.

- accept_all_cookies**
accept all cookies.
- anonymous**
apply restrictions for anonymous account, see also **-restrictions**.
- assume_charset=MIMEname**
charset for documents that do not specify it.
- assume_local_charset=MIMEname**
charset assumed for local files, i.e., files which *Lynx* creates such as internal pages for the options menu.
- assume_unrec_charset=MIMEname**
use this instead of unrecognized charsets.
- auth=ID:PASSWORD**
set authorization ID and password for protected documents at startup. Be sure to protect any script files which use this switch.
- base** prepend a request URL comment and BASE tag to text/html outputs for **-source** dumps.
- bibhost=URL**
specify a local bibp server (default `http://bibhost/`).
- blink** forces high intensity background colors for color mode, if available and supported by the terminal. This applies to the **slang** library (for a few terminal emulators), or to OS/2 EMX with **ncurses**.
- book** use the bookmark page as the startfile. The default or command line startfile is still set for the Main screen command, and will be used if the bookmark page is unavailable or blank.
- buried_news**
toggles scanning of news articles for buried references, and converts them to news links. Not recommended because email addresses enclosed in angle brackets will be converted to false news links, and uuencoded messages can be trashed.
- cache=NUMBER**
set the NUMBER of documents cached in memory. The default is 10.
- case** enable case-sensitive string searching.
- center**
Toggle center alignment in HTML TABLE.
- cfg=FILENAME**
specifies a *Lynx* configuration file other than the default `lynx.cfg`.
- child** exit on left-arrow in startfile, and disable save to disk and associated print/mail options.
- child_relaxed**
exit on left-arrow in startfile, but allow save to disk and associated print/mail options.
- cmd_log=FILENAME**
write keystroke commands and related information to the specified file.
- cmd_script=FILENAME**
read keystroke commands from the specified file. You can use the data written using the **-cmd_log** option. *Lynx* will ignore other information which the command-logging may have written to the logfile. Each line of the command script contains either a comment beginning with "#", or a keyword:
exit
causes the script to stop, and forces *Lynx* to exit immediately.

key

the character value, in printable form. Cursor and other special keys are given as names, e.g., “Down Arrow”. Printable 7-bit ASCII codes are given as-is, and hexadecimal values represent other 8-bit codes.

set

followed by a “name=value” allows one to override values set in the lynx.cfg or .lynxrc files. *Lynx* tries the cfg-file setting first.

-collapse_br_tags

toggles collapsing of BR tags.

-color forces color mode on, if available. Default color control sequences which work for many terminal types are assumed if the terminal capability description does not specify how to handle color. *Lynx* needs to be compiled with the **slang** library for this flag, it is equivalent to setting the **COLORTERM** environment variable. (If color support is instead provided by a color-capable curses library like **ncurses**, *Lynx* relies completely on the terminal description to determine whether color mode is possible, and this flag is not needed and thus unavailable.) A saved show_color=always setting found in a .lynxrc file at startup has the same effect. A saved show_color=never found in .lynxrc on startup is overridden by this flag.

-connect_timeout=N

Sets the connection timeout, where N is given in seconds.

-cookie_file=FILENAME

specifies a file to use to read cookies. If none is specified, the default value is ~/.lynx_cookies for most systems, but ~/cookies for MS-DOS.

-cookie_save_file=FILENAME

specifies a file to use to store cookies. If none is specified, the value given by **-cookie_file** is used.

-cookies

toggles handling of Set-Cookie headers.

-core toggles forced core dumps on fatal errors. Turn this option off to ask *Lynx* to force a core dump if a fatal error occurs.

-crawl with **-traversal**, output each page to a file. with **-dump**, format output as with **-traversal**, but to the standard output.

-curses_pads

toggles the use of curses “pad” feature which supports left/right scrolling of the display. The feature is normally available for curses configurations, but inactive. To activate it, use the “[” character or the LINEWRAP_TOGGLE command. Toggling this option makes the feature altogether unavailable.

-debug_partial

separate incremental display stages with MessageSecs delay

-default_colors

toggles the default-colors feature which is normally set in the lynx.cfg file.

-delay add DebugSecs delay after each progress-message

-display=DISPLAY

set the display variable for X rexec-ed programs.

-display_charset=MIMEname

set the charset for the terminal output.

-dont_wrap_pre

inhibit wrapping of text when **-dump**’ing and **-crawl**’ing, mark wrapped lines of <pre> in interactive session.

-dump

dumps the formatted output of the default document or those specified on the command line to standard output. Unlike interactive mode, all documents are processed. This can be used in the following way:

```
lynx -dump http://www.subir.com/lynx.html
```

Files specified on the command line are formatted as HTML if their names end with one of the standard web suffixes such as “.htm” or “.html”. Use the **-force_html** option to format files whose names do not follow this convention.

-editor=EDITOR

enable external editing, using the specified EDITOR. (vi, ed, emacs, etc.)

-emacskeys

enable emacs-like key movement.

-enable_scrollback

toggles compatibility with communication programs’ scrollback keys (may be incompatible with some curses packages).

-error_file=FILE

define a file where *Lynx* will report HTTP access codes.

-exec enable local program execution (normally not configured).**-fileversions**

include all versions of files in local VMS directory listings.

-find_leaks

toggle memory leak-checking. Normally this is not compiled-into your executable, but when it is, it can be disabled for a session.

-force_empty_hrefless_a

force HREF-less “A” elements to be empty (close them as soon as they are seen).

-force_html

forces the first document to be interpreted as HTML.

This is most useful when processing files specified on the command line which have an unrecognized suffix (or the suffix is associated with a non-HTML type, such as “.txt” for plain text files).

Lynx recognizes these file suffixes as HTML:

```
".ht3",
".htm",
".html3",
".html",
".htmlx",
".php3",
".php",
".phtml",
".sht", and
".shtml".
```

-force_secure

toggles forcing of the secure flag for SSL cookies.

-forms_options

toggles whether the Options Menu is key-based or form-based.

-from toggles transmissions of From headers.

- ftp** disable ftp access.
- get_data**
 properly formatted data for a *get* form are read in from the standard input and passed to the form.
 Input is terminated by a line that starts with “----”.
- Lynx* issues an HTTP **GET**, sending the form to the path or URL given on the command-line and
 prints the response of the server. If no path or URL is given, *Lynx* sends the form to the start-page.
- head** send a HEAD request for the mime headers.
- help** print the *Lynx* command syntax usage message, and exit.
- hiddenlinks=[option]**
 control the display of hidden links.
- merge**
 hidden links show up as bracketed numbers and are numbered together with other links in the
 sequence of their occurrence in the document.
- listonly**
 hidden links are shown only on **L**ist screens and listings generated by **-dump** or from the
 Print menu, but appear separately at the end of those lists. This is the default behavior.
- ignore**
 hidden links do not appear even in listings.
- historical**
 toggles use of “>” or “-->” as a terminator for comments.
- homepage=URL**
 set homepage separate from start page.
- image_links**
 toggles inclusion of links for all images.
- index=URL**
 set the default index file to the specified URL.
- ismap**
 toggles inclusion of ISMAP links when client-side MAPs are present.
- justify**
 do justification of text.
- link=NUMBER**
 starting count for lnk#.dat files produced by **-crawl**.
- list_inline**
 for **-dump**, show the links inline with the text.
- listonly**
 for **-dump**, show only the list of links.
- localhost**
 disable URLs that point to remote hosts.
- locexec**
 enable local program execution from local files only (if *Lynx* was compiled with local execution
 enabled).
- lss=FILENAME**
 specify filename containing color-style information. The default is lynx.lss. If you give an empty
 filename, *Lynx* uses a built-in monochrome scheme which imitates the non-color-style
 configuration.

- mime_header**
prints the MIME header of a fetched document along with its source.
- minimal**
toggles minimal versus valid comment parsing.
- nested_tables**
toggles nested-tables logic (for debugging).
- newschunksize=NUMBER**
number of articles in chunked news listings.
- newsmaxchunk=NUMBER**
maximum news articles in listings before chunking.
- nobold**
disable bold video-attribute.
- nobrowse**
disable directory browsing.
- nocc** disable Cc: prompts for self copies of mailings. Note that this does not disable any CCs which are incorporated within a mailto URL or form ACTION.
- nocolor**
force color mode off, overriding terminal capabilities and any **-color** flags, COLORTERM variable, and saved .lynxrc settings.
- noexec**
disable local program execution. (DEFAULT)
- nofilereferer**
disable transmissions of Referer headers for file URLs.
- nolist** disable the link list feature in dumps.
- nolog** disable mailing of error messages to document owners.
- nomargins**
disable left/right margins in the default style sheet.
- nomore**
disable **-more-** string in statusline messages.
- nonrestarting_sigwinch**
This flag is not available on all systems, *Lynx* needs to be compiled with HAVE_SIGACTION defined. If available, this flag *may* cause *Lynx* to react more immediately to window changes when run within an **xterm**.
- nonumbers**
disable link- and field-numbering. This overrides **-number_fields** and **-number_links**.
- nopause**
disable forced pauses for statusline messages.
- noprint**
disable most print functions.
- noredir**
prevents automatic redirection and prints a message with a link to the new URL.
- noreferer**
disable transmissions of Referer headers.
- noreverse**
disable reverse video-attribute.

- nosocks**
disable SOCKS proxy usage by a SOCKSified *Lynx*.
- nostatus**
disable the retrieval status messages.
- notitle**
disable title and blank line from top of page.
- nounderline**
disable underline video-attribute.
- number_fields**
force numbering of links as well as form input fields
- number_links**
force numbering of links.
- partial**
toggles display partial pages while loading.
- partial_thres=NUMBER**
number of lines to render before repainting display with partial-display logic
- passive_ftp**
toggles passive ftp connections.
- pauth=ID:PASSWORD**
set authorization ID and password for a protected proxy server at startup. Be sure to protect any script files which use this switch.
- popup**
toggles handling of single-choice SELECT options via popup windows or as lists of radio buttons.
- post_data**
properly formatted data for a *post* form are read in from the standard input and passed to the form. Input is terminated by a line that starts with "——".

Lynx issues an HTTP **POST**, sending the form to the path or URL given on the command-line and prints the response of the server. If no path or URL is given, *Lynx* sends the form to the start-page.
- preparsed**
show HTML source preparsed and reformatted when used with **-source** or in source view.
- prettysrc**
show HTML source view with lexical elements and tags in color.
- print** enable print functions. (default)
- pseudo_inlines**
toggles pseudo-ALTs for inline images with no ALT string.
- raw** toggles default setting of 8-bit character translations or CJK mode for the startup character set.
- realm**
restricts access to URLs in the starting realm.
- read_timeout=N**
Sets the read-timeout, where N is given in seconds.
- reload**
flushes the cache on a proxy server (only the first document given on the command-line is affected).
- restrictions=[option][,option][,option]...**
allows a list of services to be disabled selectively. Dashes and underscores in option names can be intermixed. The following list is printed if no options are specified.

all

restricts all options listed below.

bookmark

disallow changing the location of the bookmark file.

bookmark_exec

disallow execution links via the bookmark file.

change_exec_perms

disallow changing the eXecute permission on files (but still allow it for directories) when local file management is enabled.

default

same as command line option **-anonymous**. Disables default services for anonymous users. Set to all restricted, except for: `inside_telnet`, `outside_telnet`, `inside_ftp`, `outside_ftp`, `inside_rlogin`, `outside_rlogin`, `inside_news`, `outside_news`, `telnet_port`, `jump`, `mail`, `print`, `exec`, and `goto`. The settings for these, as well as additional goto restrictions for specific URL schemes that are also applied, are derived from definitions within `userdefs.h`.

dired_support

disallow local file management.

disk_save

disallow saving to disk in the download and print menus.

dotfiles

disallow access to, or creation of, hidden (dot) files.

download

disallow some downloaders in the download menu (does not imply `disk_save` restriction).

editor

disallow external editing.

exec

disable execution scripts.

exec_frozen

disallow the user from changing the local execution option.

externals

disallow some "EXTERNAL" configuration lines if support for passing URLs to external applications (with the `EXTERN` command) is compiled in.

file_url

disallow using **G**oto, served links or bookmarks for file: URLs.

goto

disable the "g" (goto) command.

inside_ftp

disallow ftps for people coming from inside your domain (utmp required for selectivity).

inside_news

disallow USENET news posting for people coming from inside your domain (utmp required for selectivity).

inside_rlogin

disallow rlogins for people coming from inside your domain (utmp required for selectivity).

inside_telnet

disallow telnets for people coming from inside your domain (utmp required for selectivity).

- jump**
disable the “j” (jump) command.
- multibook**
disallow multiple bookmarks.
- mail**
disallow mail.
- news_post**
disallow USENET News posting.
- options_save**
disallow saving options in .lynxrc.
- outside_ftp**
disallow ftps for people coming from outside your domain (utmp required for selectivity).
- outside_news**
disallow USENET news reading and posting for people coming from outside your domain (utmp required for selectivity). This restriction applies to “news”, “nntp”, “newspost”, and “newsreply” URLs, but not to “snews”, “snewspost”, or “snewsreply” in case they are supported.
- outside_rlogin**
disallow rlogins for people coming from outside your domain (utmp required for selectivity).
- outside_telnet**
disallow telnets for people coming from outside your domain (utmp required for selectivity).
- print**
disallow most print options.
- shell**
disallow shell escapes and lynxexec or lynxprog **G)oto**’s.
- suspend**
disallow Unix Control-Z suspends with escape to shell.
- telnet_port**
disallow specifying a port in telnet **G)oto**’s.
- useragent**
disallow modifications of the User-Agent header.
- resubmit_posts**
toggles forced resubmissions (no-cache) of forms with method POST when the documents they returned are sought with the PREV_DOC command or from the History List.
- rlogin**
disable recognition of rlogin commands.
- scrollbar**
toggles showing scrollbar.
- scrollbar_arrow**
toggles showing arrows at ends of the scrollbar.
- selective**
require .www_browsable files to browse directories.
- session=FILENAME**
resumes from specified file on startup and saves session to that file on exit.

- sessionin=FILENAME**
resumes session from specified file.
- sessionout=FILENAME**
saves session to specified file.
- short_url**
show very long URLs in the status line with “...” to represent the portion which cannot be displayed. The beginning and end of the URL are displayed, rather than suppressing the end.
- show_cfg**
Print the configuration settings, e.g., as read from “lynx.cfg”, and exit.
- show_cursor**
If enabled the cursor will not be hidden in the right hand corner but will instead be positioned at the start of the currently selected link. Show cursor is the default for systems without FANCY_CURSES capabilities. The default configuration can be changed in userdefs.h or lynx.cfg. The command line switch toggles the default.
- show_rate**
If enabled the transfer rate is shown in bytes/second. If disabled, no transfer rate is shown. Use lynx.cfg or the options menu to select KB/second and/or ETA.
- soft_dquotes**
toggles emulation of the old Netscape and Mosaic bug which treated “>” as a co-terminator for double-quotes and tags.
- source**
works the same as dump but outputs HTML source instead of formatted text. For example

```
lynx -source . >foo.html
```


generates HTML source listing the files in the current directory. Each file is marked by an HREF relative to the parent directory. Add a trailing slash to make the HREF's relative to the current directory:

```
lynx -source ./ >foo.html
```
- stack_dump**
disable SIGINT cleanup handler
- startfile_ok**
allow non-http startfile and homepage with **-validate**.
- stderr**
When dumping a document using **-dump** or **-source**, *Lynx* normally does not display alert (error) messages that you see on the screen in the status line. Use the **-stderr** option to tell *Lynx* to write these messages to the standard error.
- stdin** read the startfile from standard input (UNIX only).
- syslog=text**
information for syslog call.
- syslog_urls**
log requested URLs with syslog.
- tagsoup**
initialize parser, using Tag Soup DTD rather than SortaSGML.
- telnet** disable recognition of telnet commands.
- term=TERM**
tell *Lynx* what terminal type to assume it is talking to. (This may be useful for remote execution, when, for example, *Lynx* connects to a remote TCP/IP port that starts a script that, in turn, starts another *Lynx* process.)

- timeout=*N***
For win32, sets the network read-timeout, where *N* is given in seconds.
- tlog** toggles between using a *Lynx* Trace Log and stderr for trace output from the session.
- tna** turns on "Textfields Need Activation" mode.
- trace** turns on *Lynx* trace mode. Destination of trace output depends on **-tlog**.
- trace_mask=*value***
turn on optional traces, which may result in very large trace files. Logically OR the values to combine options:
 - 1** SGML character parsing states
 - 2** color-style
 - 4** TRST (table layout)
 - 8** configuration (lynx.cfg, .lynxrc, .lynx-keymaps, mime.types and mailcap contents)
 - 16** binary string copy/append, used in form data construction.
 - 32** cookies
 - 64** character sets
 - 128** GridText parsing
 - 256** timing
- traversal**
traverse all http links derived from startfile. When used with **-crawl**, each link that begins with the same string as startfile is output to a file, intended for indexing.
See CRAWL.announce for more information.
- trim_blank_lines**
toggles trimming of trailing blank lines as well as the related trimming of blank lines while collapsing BR tags.
- trim_input_fields**
trim input text/textarea fields in forms.
- underline_links**
toggles use of underline/bold attribute for links.
- underscore**
toggles use of `_underline_` format in dumps.
- unique_urls**
check for duplicate link numbers in each page and corresponding lists, and reuse the original link number.
- use_mouse**
turn on mouse support, if available. Clicking the left mouse button on a link traverses it. Clicking the right mouse button pops back. Click on the top line to scroll up. Click on the bottom line to scroll down. The first few positions in the top and bottom line may invoke additional functions. *Lynx* must be compiled with **ncurses** or **slang** to support this feature. If **ncurses** is used, clicking the middle mouse button pops up a simple menu. Mouse clicks may only work reliably while *Lynx* is idle waiting for input.
- useragent=*Name***
set alternate *Lynx* User-Agent header.

- validate**
accept only http URLs (for validation). Complete security restrictions also are implemented.
- verbose**
toggle [LINK], [IMAGE] and [INLINE] comments with filenames of these images.
- version**
print version information, and exit.
- vikeys**
enable vi-like key movement.
- wdebug**
enable Waterloo tcp/ip packet debug (print to watt debugfile). This applies only to DOS versions compiled with WATTCP or WATT-32.
- width=NUMBER**
number of columns for formatting of dumps, default is 80. This is limited by the number of columns that *Lynx* could display, typically 1024 (the *MAX_LINE* symbol).
- with_backspaces**
emit backspaces in output if **-dump**'ing or **-crawl**'ing (like **man** does)
- xhtml_parsing**
tells *Lynx* that it can ignore certain tags which have no content in an XHTML 1.0 document. For example "<p/>" will be discarded.

COMMANDS

More than one key can be mapped to a given command. Here are some of the most useful:

- Use **Up arrow** and **Down arrow** to scroll through hypertext links.
- **Right arrow** or **Return** will follow a highlighted hypertext link.
- **Left Arrow** or **"u"** will retreat from a link.
- Type **"H"**, **"?"**, or **F1** for online help and descriptions of key-stroke commands.
- Type **"k"** or **"K"** for a list of the current key-stroke command mappings.

If the same command is mapped to the same letter differing only by upper/lowercase only the lowercase mapping is shown.

- Type **Delete** to view history list.

ENVIRONMENT

In addition to various "standard" environment variables such as **HOME**, **PATH**, **USER**, **DISPLAY**, **TMPDIR**, etc, *Lynx* utilizes several *Lynx*-specific environment variables, if they exist.

Others may be created or modified by *Lynx* to pass data to an external program, or for other reasons. These are listed separately below.

See also the sections on **SIMULATED CGI SUPPORT** and **NATIVE LANGUAGE SUPPORT**, below.

Note: Not all environment variables apply to all types of platforms supported by *Lynx*, though most do. Feedback on platform dependencies is solicited.

Environment Variables Used By *Lynx*:

- | | |
|------------------|--|
| COLORTERM | If set, color capability for the terminal is forced on at startup time. The actual value assigned to the variable is ignored. This variable is only meaningful if <i>Lynx</i> was built using the slang screen-handling library. |
| LYNX_CFG | This variable, if set, will override the default location and name of the global configuration file (normally, lynx.cfg) that was defined by the LYNX_CFG_FILE constant in the userdefs.h file, during installation. |

See the `userdefs.h` file for more information.

| | |
|--------------------------|--|
| LYNX_CFG_PATH | If set, this variable overrides the compiled-in search-list of directories used to find the configuration files, e.g., lynx.cfg and lynx.lss . The list is delimited with ":" (or ";" for Windows) like the PATH environment variable. |
| LYNX_HELPFILE | If set, this variable overrides the compiled-in URL and configuration file URL for the <i>Lynx</i> help file. |
| LYNX_LOCALEDIR | If set, this variable overrides the compiled-in location of the locale directory which contains native language (NLS) message text. |
| LYNX_LSS | This variable, if set, specifies the location of the default <i>Lynx</i> character style sheet file. [Currently only meaningful if <i>Lynx</i> was built using curses color style support.] |
| LYNX_SAVE_SPACE | This variable, if set, will override the default path prefix for files saved to disk that is defined in the lynx.cfg SAVE_SPACE: statement. See the lynx.cfg file for more information. |
| LYNX_TEMP_SPACE | This variable, if set, will override the default path prefix for temporary files that was defined during installation, as well as any value that may be assigned to the TMPDIR variable. |
| MAIL | This variable specifies the default inbox <i>Lynx</i> will check for new mail, if such checking is enabled in the lynx.cfg file. |
| NEWS_ORGANIZATION | This variable, if set, provides the string used in the Organization: header of USENET news postings. It will override the setting of the ORGANIZATION environment variable, if it is also set (and, on UNIX , the contents of an <code>/etc/organization</code> file, if present). |
| NNTPSERVER | If set, this variable specifies the default NNTP server that will be used for USENET news reading and posting with <i>Lynx</i> , via news: URL's. |
| ORGANIZATION | This variable, if set, provides the string used in the Organization: header of USENET news postings. On UNIX , it will override the contents of an <code>/etc/organization</code> file, if present. |
| PROTOCOL_proxy | <i>Lynx</i> supports the use of proxy servers that can act as firewall gateways and caching servers. They are preferable to the older gateway servers (see WWW_access_GATEWAY , below). |

Each protocol used by *Lynx*, (http, ftp, gopher, etc), can be mapped separately by setting environment variables of the form **PROTOCOL_proxy**. Protocols are indicated in a URI by the name before ":", e.g., "http" in "http://some.server.dom:port/" for HTML.

Depending on your system configuration and supported protocols, the environment variables recognized by *lynx* may include

```
cso_proxy
finger_proxy
ftp_proxy
gopher_proxy
https_proxy
http_proxy
newspost_proxy
newsreply_proxy
news_proxy
```

```

nntp_proxy
no_proxy
snewsproxy_proxy
snewsreply_proxy
snews_proxy
wais_proxy

```

See **Lynx Users Guide** for additional details and examples.

SSL_CERT_DIR Set to the directory containing trusted certificates.

SSL_CERT_FILE Set to the full path and filename for your file of trusted certificates.

WWW_access_GATEWAY

Lynx still supports use of gateway servers, with the servers specified via “WWW_access_GATEWAY” variables (where “access” is lower case and can be “http”, “ftp”, “gopher” or “wais”). However most gateway servers have been discontinued. Note that you do not include a terminal “/” for gateways, but do for proxies specified by *PROTOCOL_proxy* environment variables.

See **Lynx Users Guide** for details.

WWW_HOME This variable, if set, will override the default startup URL specified in any of the *Lynx* configuration files.

Environment Variables **Set** or **Modified** By *Lynx*:

LYNX_PRINT_DATE This variable is set by the *Lynx* p(rint) function, to the **Date:** string seen in the document’s “**Information about**” page (= cmd), if any. It is created for use by an external program, as defined in a **lynx.cfg PRINTER:** definition statement. If the field does not exist for the document, the variable is set to a null string under **UNIX**, or “No Date” under **VMS**.

LYNX_PRINT_LASTMOD

This variable is set by the *Lynx* p(rint) function, to the **Last Mod:** string seen in the document’s “**Information about**” page (= cmd), if any. It is created for use by an external program, as defined in a **lynx.cfg PRINTER:** definition statement. If the field does not exist for the document, the variable is set to a null string under **UNIX**, or “No LastMod” under **VMS**.

LYNX_PRINT_TITLE

This variable is set by the *Lynx* p(rint) function, to the **Linkname:** string seen in the document’s “**Information about**” page (= cmd), if any. It is created for use by an external program, as defined in a **lynx.cfg PRINTER:** definition statement. If the field does not exist for the document, the variable is set to a null string under **UNIX**, or “No Title” under **VMS**.

LYNX_PRINT_URL This variable is set by the *Lynx* p(rint) function, to the **URL:** string seen in the document’s “**Information about**” page (= cmd), if any. It is created for use by an external program, as defined in a **lynx.cfg PRINTER:** definition statement. If the field does not exist for the document, the variable is set to a null string under **UNIX**, or “No URL” under **VMS**.

LYNX_TRACE If set, causes *Lynx* to write a trace file as if the **-trace** option were supplied.

LYNX_TRACE_FILE

If set, overrides the compiled-in name of the trace file, which is either **Lynx.trace** or **LY-TRACE.LOG** (the latter on the DOS/Windows platforms). The trace file is in either case relative to the home directory.

LYNX_VERSION This variable is always set by *Lynx*, and may be used by an external program to determine if it was invoked by *Lynx*.

See also the comments in the distribution's sample **mailcap** file, for notes on usage in such a file.

TERM

Normally, this variable is used by *Lynx* to determine the terminal type being used to invoke *Lynx*. If, however, it is unset at startup time (or has the value "unknown"), or if the **-term** command-line option is used (see **OPTIONS** section above), *Lynx* will set or modify its value to the user specified terminal type (for the *Lynx* execution environment). Note: If set/modified by *Lynx*, the values of the **LINES** and/or **COLUMNS** environment variables may also be changed.

SIMULATED CGI SUPPORT

If built with the **cgi-links** option enabled, *Lynx* allows access to a cgi script directly without the need for an http daemon.

When executing such "lynxcgi scripts" (if enabled), the following variables may be set for simulating a CGI environment:

CONTENT_LENGTH**CONTENT_TYPE****DOCUMENT_ROOT****HTTP_ACCEPT_CHARSET****HTTP_ACCEPT_LANGUAGE****HTTP_USER_AGENT****PATH_INFO****PATH_TRANSLATED****QUERY_STRING****REMOTE_ADDR****REMOTE_HOST****REQUEST_METHOD****SERVER_SOFTWARE**

Other environment variables are not inherited by the script, unless they are provided via a **LYNXCGI_ENVIRONMENT** statement in the configuration file. See the **lynx.cfg** file, and the (draft) CGI 1.1 Specification <<http://Web.Golux.Com/coar/cgi/draft-coar-cgi-v11-00.txt>> for the definition and usage of these variables.

The CGI Specification, and other associated documentation, should be consulted for general information on CGI script programming.

NATIVE LANGUAGE SUPPORT

If configured and installed with **Native Language Support**, *Lynx* will display status and other messages in your local language. See the file **ABOUT-NLS** in the source distribution, or at your local **GNU** site, for more information about internationalization.

The following environment variables may be used to alter default settings:

LANG This variable, if set, will override the default message language. It is an **ISO 639** two-letter code identifying the language. Language codes are **NOT** the same as the country codes given in **ISO 3166**.

LANGUAGE This variable, if set, will override the default message language. This is a **GNU** extension that has higher priority for setting the message catalog than **LANG** or **LC_ALL**.

LC_ALL and

| | |
|--------------------|--|
| LC_MESSAGES | These variables, if set, specify the notion of native language formatting style. They are POSIXly correct. |
| LINGUAS | This variable, if set prior to configuration, limits the installed languages to specific values. It is a space-separated list of two-letter codes. Currently, it is hard-coded to a wish list. |
| NLSPATH | This variable, if set, is used as the path prefix for message catalogs. |

NOTES

This is the *Lynx* v2.8.9 Release; development is in progress for 2.9.0.

If you wish to contribute to the further development of *Lynx*, subscribe to our mailing list. Send email to <lynx-dev-request@nongnu.org> with “subscribe lynx-dev” as the only line in the body of your message.

Send bug reports, comments, suggestions to <lynx-dev@nongnu.org> after subscribing.

Unsubscribe by sending email to <lynx-dev-request@nongnu.org> with “unsubscribe lynx-dev” as the only line in the body of your message. Do not send the unsubscribe message to the lynx-dev list, itself.

SEE ALSO

catgets(3), **curses(3)**, **environ(7)**, **execve(2)**, **ftp(1)**, **gettext(GNU)**, **localeconv(3)**, **ncurses(3)**, **setlocale(3)**, **slang(?)**, **termcap(5)**, **terminfo(5)**, **wget(GNU)**

Note that man page availability and section numbering is somewhat platform dependent, and may vary from the above references.

A section shown as (GNU), is intended to denote that the topic may be available via an info page, instead of a man page (i.e., try “info subject”, rather than “man subject”).

A section shown as (?) denotes that documentation on the topic exists, but is not part of an established documentation retrieval system (see the distribution files associated with the topic, or contact your System Administrator for further information).

ACKNOWLEDGMENTS

Lynx has incorporated code from a variety of sources along the way. The earliest versions of *Lynx* included code from Earl Fogel of Computing Services at the University of Saskatchewan, who implemented HYPERREZ in the Unix environment. HYPERREZ was developed by Niel Larson of Think.com and served as the model for the early versions of *Lynx*. Those versions also incorporated libraries from the Unix Gopher clients developed at the University of Minnesota, and the later versions of *Lynx* rely on the WWW client library code developed by Tim Berners-Lee and the WWW community. Also a special thanks to Foteos Macrides who ported much of *Lynx* to VMS and did or organized most of its development since the departures of Lou Montulli and Garrett Blythe from the University of Kansas in the summer of 1994 through the release of v2.7.2, and to everyone on the net who has contributed to *Lynx*’s development either directly (through patches, comments or bug reports) or indirectly (through inspiration and development of other systems).

AUTHORS

Lou Montulli, Garrett Blythe, Craig Lavender, Michael Grobe, Charles Rezac
Academic Computing Services
University of Kansas
Lawrence, Kansas 66047

Foteos Macrides
Worcester Foundation for Biomedical Research
Shrewsbury, Massachusetts 01545

Thomas E. Dickey
<dickey@invisible-island.net>

NAME

mail_file – mail file(s) as attachments

SYNOPSIS

edrc/bin/mail_file [**-h**]

echo *message* | **mail_file** **-s** "*subject*" **-t** *to* [**-c** *carbon_copy*] [**-b** *blind_carbon_copy*] [[**-d**] **-a** *attachments*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

with **mail_file** a mail with a message content and file attachments can be sent with a single command.

The message content is received from **stdin** what enables easy script integration of the command.

In the configuration file **mail_file.cfg** can be configured if the mail should be sent locally, or if the mail should be sent from a remote (hub) system. It is possible to define a list of hosts that can be used as a hub. If the local host name is part of the list, the mail is sent locally; else the mail is sent from the first host in the hub list that is up.

OPTIONS

-h usage message.

"subject" subject of the mail to be sent.

-t to comma separated list of mail recipients.

-c cc comma separated list of carbon copy mail recipients.

-b bcc comma separated list of blind carbon copy mail recipients.

-d convert attachment file(s) to DOS.

-f from sender address of the mail. The default sender address can be configured in the **mail_file.cfg** configuration file in the **DEFAULT_SENDER** setting.

-a attachments
space separated list of files to be attached to the mail.

ENVIRONMENT

-

EXIT STATUS

| | |
|-----------|--|
| 0 | no error. |
| 2 | operating system is not supported yet. See osid(3) if you get this error. |
| 3 | specified attachment file does not exist. |
| 4 | usage printed. |
| 5 | command aborted pressing Ctrl+C. |
| 6 | configfile mail_file.cfg does not exist. |
| 7 | cannot connect to mail hub with remote_shell . |
| 11 | temporary directory could not be claimed or created in /tmp . Check the system temporary directory /tmp if you get this error, it is an indicator of system intrusion. |

FILES

etc/mail_file.cfg
configuration file of **mail_file**, see **mail_file.cfg(4)** for more information.

EXAMPLES**1) send a file with a single line message:**

```
echo "See logfile in attachment" | mail_file \
-s "myApplication Logfile" \
-t "fred.flintstone@acme.com" \
-a quarry_moves.log
```

2) send a file with a multiple line message:

```
cat <<EOM | mail_file ...
Hi Fred

Please open the quarry move log in the attachment

Regards
Barney Boulder
EOM
```

SEE ALSO

edrcintro(1), **mail_file.cfg**(4), **remote_shell**(3), **sendmail**(8)

NOTES

mail_file uses the **sendmail** command internally to send the mail.

BUGS

-

AUTHOR

mail_file was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

mail_file.cfg – configuration file for mail_file

SYNOPSIS

edrc/etc/mail_file.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **mail_file** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**DEFAULT_SENDER**

Default mail sender if the **-f from** option is not specified on the command line.

Example: **DEFAULT_SENDER=fred.flintstone@acme.ch**

Default: **DEFAULT_SENDER=smc.base.services@eds.com**

MAIL_HUB

Space separated list of hosts to relay the mail sending. The relay is done using a named pipe and not using mail server functionality.

If the local host name is part of the **MAIL_HUB** list, the mail will be sent locally; else the mail is sent from the first host in the list that is up.

Example: **MAIL_HUB="acme001 acme002 acme003"**

Default: **MAIL_HUB=""**

MAIL_HUB_USER

User to connect to the mail hub as root. Other user use the own unix user id and must ensure that no password is prompted on the remote system.

Example: MAIL_HUB_USER=edrc

Default: MAIL_HUB_USER=edrc

MAIL_CLIENT

Mail client to be used. Currently supported clients are *sendmail* and *msmtp*.

Example: MAIL_CLIENT=msmtp

Default: MAIL_CLIENT=sendmail

MAIL_CLIENT_OPTS

Additional options for the chosen mail client.

Example: MAIL_CLIENT_OPTS=""

Default: MAIL_CLIENT_OPTS=""

MAIL_SERVER_NAME

Mail server. This setting is relevant only, if **MAIL_CLIENT** is set to *msmtp*.

Example: MAIL_SERVER_NAME=exchange.int.acme.ch

Default: MAIL_SERVER_NAME=mail

MAIL_SERVER_PORT

Mailserver port to connect to. This setting is relevant only, if **MAIL_CLIENT** is set to *msmtp*.

Example: MAIL_SERVER_PORT=25

Default: MAIL_SERVER_PORT=25

SEE ALSO

mail_file(1), **msmtp(3)**, **random(3)**, **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

mail_file.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

maketemp – create unique secure temporary dir

SYNOPSIS

edrc/lib/maketemp [**-h**]

maketemp [**-s** *scriptname*] [**-q**] **-d** *dir_basename*

AVAILABILITY

WA2L/edrc

DESCRIPTION

create a unique non existing temporary directory with secure permissions and return the created directory path via **stdout**.

Use **removetemp**(3) to remove a temporary directory created with **maketemp**.

Some operating systems know the **mktemp**(1) command with a very similar functionality. However, if there is a need to write cross-operating system compatible scripts, **maketemp**(3) should be used, due to the fact that the options and the behaviour of this command does not differ between the supported operating systems, as the **mktemp** command does.

OPTIONS

-h usage message.

-s *scriptname*
scriptname in message output of **maketemp**.

-q quiet, no message output.

-d *dir_basename*
directory basename.

If a temporary directory for the command **my_command** in the directory **/tmp** has to be created, **maketemp** ... **-d /tmp.my_command** has to be specified.

See section **EXAMPLES** for additional information.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|---|
| 0 | no error. |
| 1 | temporary directory could not be created. |
| 4 | usage message listed. |

FILES

<dir_basename>.<PID>.<RANDOM_STRING>/
created temporary directory.

<dir_basename>.<PID>.<RANDOM_STRING>/.maketemp.flagfile
flagfile containing the state information about the command that created the temporary directory. This file is read by the **removetemp(3)** and **listtemp(3)**.

EXAMPLES**1) create a tempdir in '/tmp':**

```
[ /root ]
[ root@acme001 ] [-sh]: maketemp -d /tmp/.example
/tmp/.example.3430.2Hpt0D/
```

2) create/use/remove a tempdir in '/tmp' in a script:

```
#!/bin/sh

Scriptname=`basename $0`
Tmp=`maketemp -s $Scriptname -d /tmp/.$Scriptname` || exit 11

:
:

cat /etc/passwd | sort > ${Tmp}sorted_passwd

:
:

removetemp -q -d ${Tmp}
```

SEE ALSO

edrcintro(1), listtemp(3), purgetemp(3), removetemp(3), mktemp(1)

NOTES

-

BUGS

-

AUTHOR

maketemp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

makeuser – user friendly interface to mkuser

SYNOPSIS

edrc/bin/makeuser [-h]

makeuser [-u *username*] [-U *uid*] [-G *groupname* | -g *gid*] [-d *home_dir*] [-c *userclass*] [-f]

AVAILABILITY

WA2L/edrc

DESCRIPTION

makeuser is a user friendly interface to the **mkuser**(3) command.

With this command a user **\$HOME** directory can be created interactively. It is not imperative that the user account is already created when creating the user **\$HOME** directory. When the user account does already exist, the defaults for all fields to be specified depend from the user name.

When entering a ? in the fields to be specified a list of available values on the system is printed. If the system is a LDAP or NIS client, the information list printed is a merge between the related name service and the information in the local files (NIS+FILES or LDAP+FILES).

All interactively queried fields can also be specified on the command line; such fields will then not be queried.

The user **\$HOME** directory is created using the **mkuser**(3) command internally.

OPTIONS

-h usage message.

-u *username*
 username of the user to create a **\$HOME** directory for.

-U *uid* User-ID of the user.

-G *groupname*
 name of the primary group the user is member of.

-g Group-ID of the primary group of the user.

-d *home_directory*
 \$HOME directory path of the user.

-c *user_class*

userclass of the user. The userclass is the template of the **\$HOME** directory. The userclass can be queried by the user using the **userclass(1)** command. To enable the **userclass(1)** command to return the userclass, a file with the name **.userclass** containing a single line with the name of the applied userclass has to be created. **makeuser(1)** also reads the **.userclass** file of the related **\$HOME** directory when re-applying the template to an already existing **\$HOME** directory.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 1** classes dir as set in the **CLASSES_DIR** setting in the **makeuser.cfg** does not exist.
- 2** **makeuser** was started with the wrong user. The start user has to be set in the **START_USER** setting in the **makeuser.cfg** file.
- 4** usage printed.
- 5** command aborted pressing Ctrl+C.
- 6** configfile **makeuser.cfg** does not exist.
- 11** temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES**etc/makeuser.cfg**

configuration file of **makeuser**, see **makeuser.cfg(4)** for more information.

\$CLASSES_DIR/index

index file for all userclasses. This file is read when displaying the available userclasses when entering ? in the "User-Class : " dialog. See **userclass.index(4)** for more information.

\$CLASSES_DIR/<userclass>/

files of a userclass.

/etc/passwd

local user account definition (NS: FILES).

/etc/group local group definition (NS: FILES).

/etc/auto.home

default local automounter map for \$HOME server location (NS: FILES). See also **makeuser.cfg(4)**.

NS: NIS nis map (**passwd**, **group**, **auto.home**).

NS: LDAP

LDAP map (**passwd** (ou=People), **group** (ou=Group), **auto.home**).

EXAMPLES

1) create a \$HOME directory for an existing user account:

```
[ / ]
[ root@host001.acme.ch ][/bin/bash]: ~edrc/bin/makeuser
makeuser - a user friendly interface to mkuser, by Chr. Walther

SPECIFY SETTINGS:

Username   : jdoe
Comment    : John Doe
User-ID (UID)   [1291]:
Group-Name   [users]:
Group-ID (GID)   : 100
Home-Directory (HOME)   [/home/jdoe]:
User-Class    : ?
  USERCLASS    COMPANY    DESCRIPTION
  -----
  adba          Customer   Application DBA User Account
  app           Customer   Application Admin User Account (Application O
  ftp-login     Customer   FTP User Account (Shell login expected)
  ftp-nologin   Customer   FTP User Account (Shell login not expected)
  user          Customer   Normal Enduser User Account
  acmedba       ACME       Database Administrator (Oracle Owner)
  acmesys       ACME       System Management Team
  monitoring    ACME       Monitoring Software System User
  notfall       ACME       Local Emergency System User
  patrol        ACME       Patrol Software System User
  root          ACME       Root System User for all Operating Systems
  _template     Template   Userclass Template, do *not* apply this userc
  (12)
User-Class   [?]: adba
Class-Description : Application DBA User Account
Force profile copy? <yn> [y] :y

SUMMARY:

Username.....: jdoe
Comment.....: John Doe
User-ID (UID).....: 1291
```

```

Group-Name.....: users
Group-ID (GID).....: 100
Home-Directory (HOME)..: /home/jdoe
User-Class.....: adba
Class-Description.....: Application DBA User Account
Force profile copy.....: y

```

```
Create User Home? <yn> [n] :y
```

```
CREATE USER $HOME DIRECTORY:
```

```

Initializing directory /home/jdoe
Created /home/jdoe/core
Created /home/jdoe/bin
Created /home/jdoe/Mail
Created /home/jdoe/.ssh
Created /home/jdoe/.dt
    transferring ACME-README
    transferring .xinitrc
    transferring .userclass
    transferring .profile
    transferring .openwin-menu
    transferring .openwin-init
    transferring .myprofile
    transferring .mylogout
    transferring .mylogin
    transferring .mylog_files
    transferring .mykshrc
    transferring .myenv
    transferring .mydtwmrc
    transferring .mycshrc
    transferring .mailrc
    transferring .logout
    transferring .login
    transferring .kshrc
    transferring .exrc
    transferring .dtprofile
    transferring .dt/dtwmrc
    transferring .defaults
    transferring .cshrc
    transferring .vimrc
    transferring .bashrc
done.

```

2) apply a different userclass to a \$HOME directory:

```

[ / ]
[ root@host001.acme.ch ][/bin/bash]: ~edrc/bin/makeuser
makeuser - a user friendly interface to mkuser, by Chr. Walther

```

```
SPECIFY SETTINGS:
```

```
Username : jdoe
```

```

Comment   : John Doe
User-ID (UID)   [1291]:
Group-Name [users]:
Group-ID (GID)   : 100
Home-Directory (HOME) [/home/jdoe]:
User-Class  [adba]: user
Class-Description : Normal Enduser User Account
Force profile copy? <yn> [y] :y

```

SUMMARY:

```

Username.....: jdoe
Comment.....: John Doe
User-ID (UID).....: 1291
Group-Name.....: users
Group-ID (GID).....: 100
Home-Directory (HOME)...: /home/jdoe
User-Class.....: user
Class-Description.....: Normal Enduser User Account
Force profile copy.....: y

```

```
Create User Home? <yn> [n] :y
```

CREATE USER \$HOME DIRECTORY:

```

Initializing directory /home/jdoe
transferring .userclass
done.

```

3) create a \$HOME directory for an non-existing user account:

```

[ / ]
[ root@host001.acme.ch ] [/bin/bash]: ~edrc/bin/makeuser
makeuser - a user friendly interface to mkuser, by Chr. Walther

```

SPECIFY SETTINGS:

```

Username   : bboulder
Comment    :
User-ID (UID)   : 1992
Group-Name    : users
Group-ID (GID)   : 100
Home-Directory (HOME) : /home/bboulder
User-Class    : user
Class-Description : Normal Enduser User Account
Force profile copy? <yn> [y] :y

```

SUMMARY:

```

Username.....: bboulder
Comment.....:
User-ID (UID).....: 1992
Group-Name.....: users
Group-ID (GID).....: 100

```



```
Home-Directory (HOME)...: /home/bboulder
User-Class.....: user
Class-Description.....: Normal Enduser User Account
Force profile copy.....: y
```

```
Create User Home? <yn> [n] :y
```

```
CREATE USER $HOME DIRECTORY:
```

```
Created /home/bboulder
Initializing directory /export/home/bboulder
Created /home/bboulder/core
Created /home/bboulder/bin
Created /home/bboulder/Mail
Created /home/bboulder/.ssh
Created /home/bboulder/.dt
    transferring ACME-README
    transferring .xinitrc
    transferring .userclass
    transferring .profile
    transferring .openwin-menu
    transferring .openwin-init
    transferring .myprofile
    transferring .mylogout
    transferring .mylogin
    transferring .mylog_files
    transferring .mykshrc
    transferring .myenv
    transferring .mydtwmrc
    transferring .mycshrc
    transferring .mailrc
    transferring .logout
    transferring .login
    transferring .kshrc
    transferring .exrc
    transferring .dtprofile
    transferring .dt/dtwmrc
    transferring .defaults
    transferring .cshrc
    transferring .bashrc
done.
```

SEE ALSO

edrcintro(1), makeuser.cfg(4), userclass.index(4), mkuser(3), passwd(4), group(4)

NOTES

-

BUGS

-

AUTHOR

makeuser was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

makeuser.cfg – configuration file for makeuser

SYNOPSIS

edrc/etc/makeuser.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **makeuser** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**CLASSES_DIR**

Classes directory where the templates are.

Example: CLASSES_DIR='homedir tools-cust'/classes

Default: CLASSES_DIR='approot'/var/makeuser/classes

START_USER

This is the user makeuser has to be started with.

Example: START_USER=root

Default: START_USER=root

LOG

Log output dir of makeuser. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: LOG='homedir ACME'/log

Default: LOG=var/log

AUTO_HOME

auto home file in **/etc/** or auto home NIS map name

Example: AUTO_HOME=auto_home

Default: AUTO_HOME=auto.home

SEE ALSO

makeuser(1), **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

makeuser.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

man2html – format a manual page in html

SYNOPSIS

man2html [options] [file]

DESCRIPTION

man2html converts a manual page as found in *file* (or stdin, in case no file argument, or the argument "-", is given) from man-style nroff into html, and prints the result on stdout. It does support tbl but does not know about eqn. The exit status is 0. If something goes wrong, an error page is printed on stdout.

This can be used as a stand-alone utility, but is mainly intended as an auxiliary, to enable users to browse their man pages using a html browser like **lynx**(1), **xmosaic**(1) or **netscape**(1).

The main part of **man2html** is the troff-to-html engine written by Richard Verhoeven (rcb5@win.tue.nl). It adds hyperlinks for the following constructs:

| | |
|-----------------|--|
| foo(3x) | "http://localhost/cgi-bin/man/man2html?3x+foo" |
| method://string | "method://string" |
| www.host.name | "http://www.host.name" |
| ftp.host.name | "ftp://ftp.host.name" |
| name@host | "mailto:name@host" |
| <string.h> | "file:/usr/include/string.h" |

(The first of these can be tuned by options - see below.) No lookup is done - the links generated need not exist. Also an index with internal hyperlinks to the various sections is generated, so that it is easier to find one's way in large man pages like **bash**(1).

OPTIONS

When reading from stdin, it is not always clear how to do .so expansion. The **-D** option allows a script to define the working directory.

-D pathname

Strip the last two parts from the pathname, and do a *chdir(dir)* before starting the conversion.

The **-E** option allows the easy generation of error messages from a cgi script.

-E string

Output an error page containing the given error message.

The general form of a hyperlink generated for a man page reference is

<method:cgi><man2htmlpath><separator><manpage>

with a default as shown above. The parts of this hyperlink are set using the various options.

-h Set method:cgi to http://localhost. This is the default.

-H *host[.domain][:port]*

Set method:cgi to http://*host.domain:port*.

-l Set method:cgi to lynxcgi:/home/httpd.

-L *dir* Set method:cgi to lynxcgi:*dir*.

-M *man2htmlpath*

Set the man2htmlpath to use. The default is /cgi-bin/man/man2html.

-p Set separator to '/'.

-q Set separator to '?'. This is the default.

-r Use relative html paths, instead of cgi-bin paths.

On a machine without running **httpd**, one can use **lynx** to browse the man pages, using the lynxcgi method.

When some http daemon is running, lynx, or any other browser, can be used to browse the man pages, using the http method. The option `-l` (for 'lynxcgi') selects the former behaviour. With it, the default cgi path is */home/httpd*.

In general, a cgi script can be called by

`<path_to_script>/<more_path>?<query>`

and the environment variables `PATH_INFO` and `QUERY_STRING` will be set to `<more_path>` and `<query>`, respectively. Since lynxcgi does not handle the `PATH_INFO` part, we generate hyperlinks with '?' as a separator by default. The option `-p` (for 'path') selects '/' as a separator, while the option `-q` (for 'query') selects '?' as a separator.

The option `-H host` will specify the host to use (instead of *localhost*). A cgi script could use

`man2html -H $SERVER_NAME`

if the variable `SERVER_NAME` is set. This would allow your machine to act as a server and export man pages.

BUGS

There are many heuristics. The output will not always be perfect. The lynxcgi method will not work if lynx was compiled without selecting support for it. There may be problems with security.

AUTHOR

Richard Verhoeven was the original author of **man2html**. Michael Hamilton and Andries Brouwer subsequently improved on it. Federico Lucifredi <flucifredi@acm.org> is the current maintainer.

SEE ALSO

lynx(1), **man(1)**

NAME

mandir – directory of manual page online resources

SYNOPSIS

-

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is a directory of online resources of manual pages for different operating systems:

Multiple

| OS | VERSION | URL | DATE |
|----------|---------|---|------------|
| Multiple | | https://nixdoc.net/man-pages | 2020-12-06 |
| | | https://www.unix.com/man-page-repository.php | 2020-12-06 |
| | | http://www.polarhome.com/service/man/ | 2020-12-06 |
| | | https://man.freebsd.org/ | 2025-03-13 |

Linux

| OS | VERSION | URL | DATE |
|----------|--|--|--|
| Linux | | https://man7.org/linux/man-pages/dir_all_alphabetic.html https://nixdoc.net/man-pages/Linux/ https://manned.org/ https://man.freebsd.org/ | 2020-12-06 2020-12-06 2025-03-01 2025-03-13 |
| CentOS | * * * 7.0 | http://www.polarhome.com/service/man/?of=CentOS https://man.freebsd.org/cgi/man.cgi?manpath=centos https://manned.org/man/centos/intro https://www.unix.com/man-page-centos-repository.php | 2020-12-06 2025-03-13 2025-03-01 2020-12-06 |
| Debian | * * * * 7.7 | https://man.cx/ https://manpages.debian.org https://manned.org/man/debian/intro https://man.freebsd.org/cgi/man.cgi?manpath=debian https://www.unix.com/man-page-debian-repository.php | 2021-04-22 2020-12-06 2025-03-01 2025-03-13 2020-12-06 |
| Fedora | * * | http://www.polarhome.com/service/man/?of=Fedora https://manned.org/man/fedora/intro | 2020-12-06 2025-03-01 |
| FreeBSD | * * * * 11.0 | https://www.freebsd.org/cgi/man.cgi https://nixdoc.net/man-pages/FreeBSD/ https://man.freebsd.org/cgi/man.cgi?manpath=freebsd https://manned.org/man/freebsd/intro https://www.unix.com/man-page-freebsd-repository.php | 2020-12-06 2020-12-06 2025-03-13 2025-03-01 2020-12-06 |
| OpenBSD | * * | https://nixdoc.net/man-pages/OpenBSD/ https://man.freebsd.org/cgi/man.cgi?manpath=openbsd | 2020-12-06 2025-03-13 |
| OpenSuSE | * * * 11.3 | http://www.polarhome.com/service/man/?of=OpenSuSE https://man.freebsd.org/cgi/man.cgi?manpath=opensuse https://manned.org/man/openbsd/intro https://www.unix.com/man-page-suse-repository.php | 2020-12-06 2025-03-13 2025-03-01 2020-12-06 |
| RedHat | 9 (i386) * | https://www.unix.com/man-page-redhat-repository.php https://man.freebsd.org/cgi/man.cgi?manpath=Red+Hat+9.0 | 2020-12-06 2025-03-13 |
| Ubuntu | * * 22.04 21.10 20.04 18.04 | http://manpages.ubuntu.com/cgi-bin/search.py?q=man https://man.freebsd.org/cgi/man.cgi?manpath=ubuntu https://manpages.ubuntu.com/manpages/jammy/en/ https://manpages.ubuntu.com/manpages/impish/en/ https://manpages.ubuntu.com/manpages/focal/en/ https://manpages.ubuntu.com/manpages/bionic/en/ | 2020-12-06 2025-03-13 2022-06-26 2022-06-26 2022-06-26 2022-06-26 |

UNIX

| OS | VERSION | URL | DATE |
|-------------|---------|---|------------|
| HP-UX | 11i | https://docstore.mik.ua/manuals/hp-ux/en/B2355-60130/ | 2020-12-06 |
| | 11i | https://nixdoc.net/man-pages/HP-UX/ | 2020-12-06 |
| | 10.31 | https://www.unix.com/man-page-hpux-repository.php | 2020-12-06 |
| | * | https://man.freebsd.org/cgi/man.cgi?manpath=hpux | 2025-03-13 |
| Solaris | * | https://docs.oracle.com/en/operating-systems/solaris.html | 2020-12-06 |
| | * | https://man.freebsd.org/cgi/man.cgi?manpath=solaris | 2025-03-13 |
| | 11.4 | https://docs.oracle.com/cd/E88353_01/#group-12 | 2020-12-06 |
| | 11.3 | https://docs.oracle.com/cd/E53394_01/#group-13 | 2020-12-06 |
| | 11.2 | https://docs.oracle.com/cd/E36784_01/#group-12 | 2020-12-06 |
| | 11.1 | https://docs.oracle.com/cd/E26502_01/#group-12 | 2020-12-06 |
| | 11 | https://docs.oracle.com/cd/E23824_01/#giptgh | 2020-12-06 |
| | 10 | https://docs.oracle.com/cd/E26505_01/#group-9 | 2020-12-06 |
| | 10 | https://www.unix.com/man-page-sunos-repository.php | 2020-12-06 |
| | 9 | https://docs.oracle.com/cd/E19683-01/ | 2020-12-06 |
| | 9 | https://shrubby.net/solaris9ab/SUNWaman/ | 2020-12-06 |
| | 8 | https://docs.oracle.com/cd/E19455-01/ | 2020-12-06 |
| | 7 | https://docs.oracle.com/cd/E19620-01/ | 2020-12-06 |
| | 2.6 | https://docs.oracle.com/cd/E19504-01/ | 2020-12-06 |
| | 2.5.1 | https://docs.oracle.com/cd/E19695-01/ | 2020-12-06 |
| | 2.5 | https://docs.oracle.com/cd/E19641-01/ | 2020-12-06 |
| | 2.4 | https://docs.oracle.com/cd/E19457-01/ | 2020-12-06 |
| OpenSolaris | 2009.6 | https://www.unix.com/man-page-opensolaris-repository.php | 2020-12-06 |
| Illumos | * | https://illumos.org/man/ | 2022-06-26 |
| AIX | 7.3 | https://www.ibm.com/docs/en/aix/7.3?topic=commands | 2022-06-26 |
| | 7.2 | https://www.ibm.com/docs/en/aix/7.2?topic=commands | 2020-12-14 |
| | 7.1 | https://www.ibm.com/docs/en/aix/7.1?topic=commands | 2020-12-14 |

FILEFORMAT

-

OPTIONS

-

EXAMPLES

-

SEE ALSO**edrcintro(1), edrcman(1), man(1), manpages(4), osid(3), osid.dat(4)****NOTES**

Be aware that many of the manpages listed above can be queried directly from the terminal using the '**edrc-man -i system manpage**' and/or '**edrcman -o os manpage**' command.

BUGS

-

AUTHOR

mandir was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

manpages – writing manual pages for WA2L/edrc

SYNOPSIS

-

AVAILABILITY

WA2L/edrc

DESCRIPTION

This man page gives an overview of used **man** macros, the general structure and some conventions of the manual pages for WA2L/edrc.

MANPAGE ORGANIZATION

Due to the fact that WA2L/edrc runs on multiple operating systems some man pages differ for some different operating systems. This is especially the case for compiled commands distributed with WA2L/edrc. Those operating system dependent man pages are marked with a plus (+) in the **EDRC(1)** man page. Place operating system dependent manual pages to the correct directory related to the osid in the **man/** basedir. See **edrcman(1)** for the description of the placement of the man pages.

MANPAGE SECTIONS

Unfortunately the man page sections are organized differently in different operating systems.

The man pages in WA2L/edrc are organized in the following sections which is very similar to the SunOS operating system:

- | | |
|--------------|--|
| man1 | General Commands |
| man1m | Maintenance (administrative) Commands |
| man3 | Library Functions and Library Commands |
| man4 | File Formats and Configuration Files |

MANPAGE STRUCTURE

All listed sections below should be used when writing a man page for the WA2L/edrc package. If there are no information for a certain section, do not remove the section title, simply add a dash (–) instead of the text. This shows to the reader that the author of the man page did not forget the section, but there is

currently no information needed there.

MANPAGES FOR COMMANDS

NAME name and short description of command.

SYNOPSIS

usage of the command.

AVAILABILITY

where the command is available (WA2L/edrc).

DESCRIPTION

detail command description.

OPTIONS

command line options.

ENVIRONMENT

environment variables that influence the command.

EXIT STATUS

all exit states of the command.

FILES

used files by the command.

EXAMPLES

examples of command usage.

SEE ALSO

reference to other man pages or documents.

NOTES

additional command notes that add additional information not given in the DESCRIPTION section.

BUGS

known bugs or limitations of the command.

AUTHOR

author(s) of the command and hint where to send bug reports and suggestions.

COPYRIGHT

copyright notice.

MANPAGES FOR CONFIGURATION FILES

NAME name and short description of the file.

SYNOPSIS

file name and location.

AVAILABILITY

where the command is available (WA2L/edrc).

DESCRIPTION

detail file description.

FILEFORMAT

file format description.

OPTIONS

detail description of all configuration file settings (option,value pairs).

EXAMPLES

examples of the config file.

SEE ALSO

reference to other man pages or documents.

NOTES

additional command notes that add additional information not given in the DESCRIPTION section.

BUGS

known bugs or limitations of the command.

AUTHOR

author(s) of the command and hint where to send bug reports and suggestions.

COPYRIGHT

copyright notice.

CONVENTIONS FOLLOWED IN WA2L/edrc MAN PAGES

The following conventions were followed while writing the man pages for the WA2L/edrc package:

files

are set in bold face (.B).

directories

are set in bold face (.B).

indentation steps

the indentation for all sections, where labels are used is 10.

command line options

are set in bold face (.B).

configuration file options

are set in bold face (.B).

command line settings

are set in italic (.I).

configuration settings (=values)

are set in italic (.I).

using of foreign inventions

if parts of code, documentation or concepts from other individuals as the principal author of the command, config file or manual page is used those individuals should be mentioned in the NOTES and/or AUTHOR section of the man page.

references in SEE ALSO

each manual page should at least reference the **edrcintro**(1) man page.

edrcintro man page

each command that is distributed with WA2L/edrc must have a short description in the **edrcintro**(1) man page.

EDRC man page

each command and related config file that is distributed with WA2L/edrc must have an entry in the **EDRC**(1) man page, this is the main index to all commands and config files.

PostScript, PDF, HTML and eBook man pages

the man pages are formatted to PostScript, PDF, HTML and eBooks using the Makefiles located in **doc/.man/**. To generate the PostScript, PDF and HTML man pages invoke: **cd ~edrc/doc/.man ; make**. If a new man page is written or included to the WA2L/edrc package the **doc/.man/Makefile** has to be edited.

USED MANUAL PAGE MACROS WITH EXAMPLES

The list of macros available for man page exceeds the one given below by many more possibilities. The macros here are those used in the WA2L/edrc package and the explained usage is related to this special usage case. Therefore the option description is given for this particular usage. See **groff_man**(7) for a more complete description.

.TH Set the title of the man page.

Format:

```
.TH manpage section "day month year" "PACKAGE" "SECTION"
```

Example:

```
.TH watchdog 1 "28 November 2005" "EDRC" "General Commands"
```

Result:

```
watchdog(1)          General Commands          watchdog(1)
:
:
```

EDRC

28 November 2005

watchdog(1)

.SH Sets up an unnumbered section heading sticking out to the left.

Format:

`.SH TEXT`

Example:

`.SH AVAILABILITY`

Result:

AVAILABILITY

.SS Sets up a secondary, unnumbered section heading.

Format:

`.SS TEXT`

Example:

`.SS GLOBAL OPTIONS`

Result:

GLOBAL OPTIONS

.TP Sets up an indented paragraph list with label. The indentation is set to `nnn` if that argument is supplied (the default unit is 'n' if omitted), otherwise it is set to the default indentation value.

The first line of text following this macro is interpreted as a string to be printed flush-left, as it is appropriate for a label. It is not interpreted as part of a paragraph, so there is no attempt to fill the first line with text from the following input lines.

Nevertheless, if the label is not as wide as the indentation, then the paragraph starts at the same line (but indented), continuing on the following lines. If the label is wider than the indentation, then the descriptive part of the paragraph begins on the line following the label, entirely indented.

Format:

`.TP indent
.B label_1
description text`

`.TP
.B label_2
description text`

Example:

`.TP 10
.B \-h
usage message.`

```
.TP
.BI "\-n " name
name of the watchdog.
```

Result:

```
-h          usage message.

-n name     name of the watchdog.
```

.IP Sets up a indented list with a marker.

Format:

```
.IP marker [ width ]
description text

.IP marker
description text
```

Example:

available colors are:

```
.IP - 2
red
```

```
.IP -
green
```

```
.IP -
yellow
```

Result:

available colors are:

```
- red

- green

- yellow
```

.PP End an indented paragraph list.

This is only needed, when additional text has to be set in the same section after the indented list.

Format:

```
.TP indent
.B label_1
description text

.TP
```



```
.B label_2
description text
.PP

non indented text
```

Example:

```
.TP 10
.B Mauna Kea
Shield volcano in Hawaii.

.TP
.B Parinacota
Strato volcano in Chile near the border to Bolivia.
.PP

For a list of all volcanos, see:
.B https://www.volcanodiscovery.com/de/volcanoes/alphabetical-list/a-z.
```

Result:

```
Mauna Kea
    Shield volcano in Hawaii.

Parinacota
    Strato volcano in Chile.
```

For a list of all volcanos, see: **<https://www.volcanodiscovery.com/de/volcanoes/alphabetical-list/a-z.html>**

.RS This macro moves the left margin to the right by the value *nnn* if specified (default unit is 'n'; otherwise the default indentation value is used. Calls to the RS macro can be nested.

Format:

```
.RS +number
```

Example:

```
.RS +10
This text is moved by 10 characters to the right.
.RE
```

Result:

```
    This text is moved by 10 characters to the right.
```

.RE This macro moves the left margin back to level *nnn*; if no argument is given, it moves one level back. The first level (i.e., no call to RS yet) has number 1, and each call to RS increases the level by 1.

Format:

```
.RE
```

Examples:

```
.RS +10
This text is moved by 10 characters to the right.
.RS +5
This text is moved 5 more characters to the right.
.RE
.RE
```

Result:

```
    This text is moved by 10 characters to the right.
        This text is moved 5 more characters to the right.
```

- .B** Causes text to appear in bold face. If no text is present on the line where the macro is called, then the text of the next line appears in bold face.

Format:

```
.B text
```

Example:

```
.B abort
```

Result:

```
abort
```

- .I** Causes text to appear in italic. If no text is present on the line where the macro is called, then the text of the next line appears in italic.

Format:

```
.I italic text
```

Example:

```
.I check_script
```

Result:

```
check_script
```

- .BI** Causes text on the same line to appear alternately in bold face and italic. The text must be on the same line as the macro call.

Format:

```
.BI bold italic
```

Example:

```
.BI "-c " check_script
```

Result:

```
-c check_script
```

.IB Causes text on the same line to appear alternately in italic and bold face. The text must be on the same line as the macro call.

Format:

`.IB italic bold`

Example:

`.IB filename ".txt"`

Result:

*filename***.txt**

.BR Causes text on the same line to appear alternately in bold face and roman. The text must be on the same line as the macro call.

Format:

`.BR bold roman`

Example:

`.BR edrcintro (1)`

Result:

edrcintro(1)

.RB Causes text on the same line to appear alternately in roman and bold face. The text must be on the same line as the macro call.

Format:

`.RB roman bold`

Example:

`.BR (/etc/hosts)`

Result:

(etc/hosts)

USED TROFF COMMANDS WITH EXAMPLES

This are commands and escape sequences that were used in the WA2L/edrc manual pages and have been proofed to be compatible over various platforms and understood by **troff**(1) and the GNU variant **groff**(1). There are many more troff/groff commands available on the various Unix and Linux dialects as the ones explained here. For more information see **troff**(1), **groff**(1) and **groff_char**(7).

\(space)

Set a space. This is useful if after a .SS an empty line should be printed.

Format:

`\`

Example:

```
.SS "Step 5: Configure logcheckd"
\

The daemon is configured editing ...
```

\&. Set a dot (.). This is useful if you have to set a dot at the beginning of a line.

Format:

```
\&.
```

Example:

```
the check specified in option
.B -c
\&. If the
```

Result:

```
the check specified in option -c. If the
```

.ds Define a shortcut that can be used in the whole manpage.

Format:

```
.ds name content
```

Example:

```
.ds PK WA2L/edrc
```

```
This is the install directory of the \*(PK package.
```

Result:

```
This is the install directory of the WA2L/edrc package.
```

.\" Comment in a man page.

Format:

```
.\" comment text
```

Example:

```
.\"
.\" manpages.4 - man page for writing manual pages
.\"
.\" [00] 17.06.2006 CWa      Initial Version
.\"
```

.nf The following lines are not formatted and the output is made verbatim as typed. However, short-cuts are resolved.

Format:

```
.nf
```

```
verbatim text
.fi
```

Example:

```
.nf
#
# /etc/hosts - internet hosts list
#
# [00] 01.01.2000 CWa   Initial Version
#
127.0.0.1      localhost
192.168.75.1   w2mzv7t001
192.168.75.129 fc4mzv7t001
.fi
```

.fi End of verbatim output.

Format:

```
.nf
verbatim text
.fi
```

Example:

```
.nf
#
# /etc/hosts - internet hosts list
#
# [00] 01.01.2000 CWa   Initial Version
#
127.0.0.1      localhost
192.168.75.1   w2mzv7t001
192.168.75.129 fc4mzv7t001
.fi
```

.br Insert a line break.

Format:

```
.br
```

Example:

```
This is the first,
.br
that the second and
.br
this is the third line.
```

Result:

```
This is the first,
that the second and
this is the third line.
```

.bp Insert a page break in PostScript/PDF output.

Format:

.bp

Example:

This is on one page,
.bp
And this is on the next page.

' Insert a single quote (').

Format:

'

Example:

.BI MESSAGE=' Text '

Result:

MESSAGE='Text'

" Insert a formatted double quote (").

Format:

.BI "bold=\[dq]" underlined "\[dq]"

Example:

.BI "MESSAGE=\[dq]" Text "\[dq]"

Result:

MESSAGE="Text"

**** Insert a backslash (\).

Format:

\e

Example:

find /data -depth -print \e
cpio -dtvum /save

Result:

find /data -depth -print \
cpio -dtvum /save

\(co Insert the copyright symbol (©).

Format:

\(co

Example:

```
Copyright \ (co 2009 by ACME Switzerland
```

Result:

```
Copyright © 2009 by ACME Switzerland
```

\[tm] Insert the trade mark symbol (™).

Format:

```
\[tm]
```

Example:

```
Trade Mark \[tm] 2009 by ACME Switzerland
```

Result:

```
Trade Mark ™ 2009 by ACME Switzerland
```

.ft CW Switch font to Courier from current line forward. This is seen in a PostScript, PDF or HTML output only.

Format:

```
.ft CW
Text in Courier
.ft R
Text in Times Roman
```

Example:

```
.ft CW
ls -lisa *.txt
.ft R
```

.ft R Switch font to Times Roman from current line forward. This is seen in a PostScript, PDF or HTML output only.

Format:

```
.ft CW
Text in Courier
.ft R
Text in Times Roman
```

Example:

```
.ft CW
ls -lisa *.txt
.ft R
```

\fB Switch font to bold from within a line forward.

- \fI** Switch font to italic from within a line forward.
- \fC** Switch font to Courier from within a line forward.
- \f(CW** Switch font to Courier from within a line forward.
- \fR** Switch font to Times Roman from within a line forward.
- \fP** Switch font to previous font setting from within a line forward.

Format:

`\fB`

Example:

`\fBcp \fR[\fIoption\fR] \fIsource dest`

Result:

cp [*option*] *source dest*

USEFUL OWN DEFINED MACROS

- *(GO** print an "=>" arrow on the terminal and a typographic "->" arrow in a PostScript file.

Definition:

```
.\" print => on terminal, -> on PostScript
.if n .ds GO =>
.if t .ds GO \(->
..
```

Format:

`*(GO text`

Example:

`*(GO goto step 1.`

Result:

→ goto step 1.

- *(co** print an "(c)" copyright on the terminal and a typographic "(c)" copyright symbol, a c in a circle, in a PostScript file.

Definition:

```
.\" print (c) on terminal, c in circle on PostScript
.if n .ds co (c)
.if t .ds co \(\co
..
```

Format:

Copyright `*(co` by ACME Switzerland

Example:

```
Copyright \*(co by ACME Switzerland
```

Result:

```
Copyright © by ACME Switzerland
```

.VB .VE Verbatim Begin, Verbatim End. Set the text between **.VB** and **.VE** verbatim and in Courier font.

Definition:

```
.de VB \" verbatim begin
.ft CW
.nf
..
```

```
.de VE \" verbatim end
.ft R
.fi
..
```

Format:

```
.VB
verbatim text set in courier
.VE
```

Example:

```
.VB
#
# /etc/hosts - internet hosts list
#
# [00] 01.01.2000 CWa    Initial Version
#
127.0.0.1      localhost
192.168.75.1   w2mzv7t001
192.168.75.129 fc4mzv7t001
.VE
```

Result:

```
#
# /etc/hosts - internet hosts list
#
# [00] 01.01.2000 CWa    Initial Version
#
127.0.0.1      localhost
192.168.75.1   w2mzv7t001
192.168.75.129 fc4mzv7t001
```

.LS .LI .LE

List Start, List Item, List End. Create a unnumbered indented list with list items preceded by a dash (-).

Definition:

```
.de LS \" List Start
.PD 0
.RS 10
..

.de LI \" List Item
.IP - 2
..

.de LE \" List End
.RE
.PP
..
```

Format:

```
.LS
.LI
first list item
.LI
another list item
.LI
one more list item
.LE
```

Example:

```
.LS
.LI
Bern is the capital of Switzerland.
.LI
Berlin is the capital of Germany.
.LI
Paris is the capital of France.
.LI
Rome is the capital of Italy.
.LE
```

Result:

- Bern is the capital of Switzerland.
- Berlin is the capital of Germany.
- Paris is the capital of France.
- Rome is the capital of Italy.

SEE ALSO

edrcintro(1), EDRC(1), edrc(1), edrcman(1), man(1), mandir(4), troff(1), groff(1), groff_char(7), groff_man(7), man(7), https://www.gnu.org/software/groff/manual/html_node/index.html#toc-Macro-Packages-2

NOTES

Some of the macro descriptions were extracted from the **groff_man(7)** man page for Groff Version 1.17.2, written by Werner Lemberg <wl@gnu.org> that is distributed with RedHat Linux 7.2.

BUGS

-

AUTHOR

manpages was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

manvi – edit manual pages in vi

SYNOPSIS

edrc/bin/manvi *manpage*

AVAILABILITY

WA2L/edrc

DESCRIPTION

with **manvi** manual pages located in the **man/*/*/** directories and additional locations as defined in the **etc/manvi.cfg** configuration file can be conveniently edited in the **vi(1)** editor without the need to browse to the file location.

Prior to the editing the manual page file is saved automatically using the **vsav(1)** command internally.

See **man(1)** for additional information.

OPTIONS

manpage manual page to be edited.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

man/*/*/ directories of WA2L/edrc manual pages.

etc/manvi.cfg
configuration file for **manvi(1)** containing additional manual page locations.

EXAMPLES

-

SEE ALSO

edrcintro(1), **man(1)**, **man.cfg(4)**, **manvi.cfg(4)**, **vsav(1)**, **vi(1)**

NOTES

-

BUGS

-

AUTHOR

manvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/man/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

manvi.cfg – configuration file for manvi

SYNOPSIS

edrc/etc/manvi.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **.manvi** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**EDITMAN_PATH**

Additional path to search for man page files to be edited.

Example: EDITMAN_PATH="homedir fred/var/Dropbox/bin/WA2L*/man"

Default: EDITMAN_PATH=

SEE ALSO

edrcintro(1), edrcman.cfg(1), manvi(1)

NOTES

-

BUGS

-

AUTHOR

manvi.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

mc – Visual shell for Unix-like systems.

SYNOPSIS

mc [-abcCdFhPstuUVx] [-l log] [dir1 [dir2]] [-e [file] ...] [-v file]

DESCRIPTION

GNU Midnight Commander is a directory browser/file manager for Unix-like operating systems.

OPTIONS

-a, --stickchars

Disable usage of graphic characters for line drawing.

-b, --nocolor

Force black and white display.

-c, --color

Force color mode, please check the section Colors for more information.

-C arg, --colors=arg

Specify a different color set in the command line. The format of arg is documented in the Colors section.

--configure-options

Display configure options.

-d, --nomouse

Disable mouse support.

-D N, --debuglevel=N

Save the debug level for SMB VFS. N is in 0–10 range.

-e [file], --edit[=file]

Start the internal editor. If the file is specified, open it on startup. See also **mcedit (1)**.

-f, --datadir

Display the compiled-in search paths for Midnight Commander files.

-F, --datadir-info

Display extended info about compiled-in paths for Midnight Commander.

-g, --oldmouse

Force a "normal tracking" mouse mode. Used when running on xterm-capable terminals (tmux/screen).

-k, --resetsoft

Reset softkeys to their default from the termcap/terminfo database. Only useful on HP terminals when the function keys don't work.

-K file, --keymap=file

Specify a name of keymap file in the command line.

-l file, --ftplog=file

Save the ftpfs dialog with the server in file.

--nokeymap

Don't load key bindings from any file, use default hardcoded keys.

-P file, --printwd=file

Print the last working directory to the specified file. This option is not meant to be used directly. Instead, it's used from a special shell script that automatically changes the current directory of the shell to the last directory Midnight Commander was in. Source the file **/usr/local/libexec/mc/mc.sh** (bash and zsh users) or **/usr/local/libexec/mc.csh** (tcsh users) respectively to define **mc** as an alias to the appropriate shell script.

- `-s, --slow`
Turn on the slow terminal mode, in this mode the program will not draw expensive line drawing characters and will toggle verbose mode off.
- `-S arg, --skin=arg`
Specify a name of skin in the command line. Technology of skins is documented in the Skins section.
- `-t, --termcap`
Used only if the code was compiled with S-Lang and terminfo: it makes Midnight Commander use the value of the **TERMCAP** variable for the terminal information instead of the information on the system wide terminal database
- `-u, --nosubshell`
Disable use of the concurrent shell (only makes sense if Midnight Commander has been built with concurrent shell support).
- `-U, --subshell`
Enable use of the concurrent shell support (only makes sense if the Midnight Commander was built with the subshell support set as an optional feature).
- `-v file, --view=file`
Start the internal viewer to view the specified file. See also **mcview (1)**.
- `-V, --version`
Display the version of the program.
- `-x, --xterm`
Force xterm mode. Used when running on xterm-capable terminals (two screen modes, and able to send mouse escape sequences).
- `-X, --no-x11`
Do not use X11 to get the state of modifiers Alt, Ctrl, Shift

If both paths are specified, the first path name is the directory to show in the active panel; the second path name is the directory to be shown in the other panel.

If one path is specified, the path name is the directory to show in the active panel; value of "other_dir" from panels.ini is the directory to be shown in the passive panel.

If no paths are specified, current directory is shown in the active panel; value of "other_dir" from panels.ini is the directory to be shown in the passive panel.

Overview

The screen of Midnight Commander is divided into four parts. Almost all of the screen space is taken up by two directory panels. By default, the second line from the bottom of the screen is the shell command line, and the bottom line shows the function key labels. The topmost line is the menu bar line. The menu bar line may not be visible, but appears if you click the topmost line with the mouse or press the F9 key.

Midnight Commander provides a view of two directories at the same time. One of the panels is the current panel (a selection bar is in the current panel). Almost all operations take place on the current panel. Some file operations like Rename and Copy by default use the directory of the unselected panel as a destination (don't worry, they always ask you for confirmation first). For more information, see the sections on the Directory Panels, the Left and Right Menus and the File Menu.

You can execute system commands from Midnight Commander by simply typing them. Everything you type will appear on the shell command line, and when you press Enter, Midnight Commander will execute the command line you typed; read the Shell Command Line and Input Line Keys sections to learn more about the command line.

Mouse Support

Midnight Commander comes with mouse support. It is activated whenever you are running on an **xterm(1)** terminal (it even works if you take a telnet, ssh or rlogin connection to another machine from the xterm) or

if you are running on a Linux console and have the **gpm** mouse server running.

When you left click on a file in the directory panels, that file is selected; if you click with the right button, the file is marked (or unmarked, depending on the previous state).

Double-clicking on a file will try to execute the command if it is an executable program; and if the extension file has a program specified for the file's extension, the specified program is executed.

Also, it is possible to execute the commands assigned to the function key labels by clicking on them.

The default auto repeat rate for the mouse buttons is 400 milliseconds. This may be changed to other values by editing the `~/config/mc/ini` file and changing the `mouse_repeat_rate` parameter.

If you are running Midnight Commander with the mouse support, you can get the default mouse behavior (cutting and pasting text) by holding down the Shift key.

Keys

Some commands in Midnight Commander involve the use of the *Control* (sometimes labeled CTRL or CTL) and the *Meta* (sometimes labeled ALT or even Compose) keys. In this manual we will use the following abbreviations:

C-**<chr>**

means hold the Control key while typing the character **<chr>**. Thus C-f would be: hold the Control key and type f.

Alt-**<chr>**

means hold the Meta or Alt key down while typing **<chr>**. If there is no Meta or Alt key, type *Esc*, release it, then type the character **<chr>**.

S-**<chr>**

means hold the Shift key down while typing **<chr>**.

All input lines in Midnight Commander use an approximation to the GNU Emacs editor's key bindings (default).

You may redefine key bindings. See *redefine hotkey bindings*

for more info. All other key bindings (described in this manual) are relative to default behavior.

There are many sections which tell about the keys. The following are the most important.

The File Menu section documents the keyboard shortcuts for the commands appearing in the File menu. This section includes the function keys. Most of these commands perform some action, usually on the selected file or the tagged files.

The Directory Panels section documents the keys which select a file or tag files as a target for a later action (the action is usually one from the file menu).

The Shell Command Line section list the keys which are used for entering and editing command lines. Most of these copy file names and such from the directory panels to the command line (to avoid excessive typing) or access the command line history.

Input Line Keys are used for editing input lines. This means both the command line and the input lines in the query dialogs.

Redefine hotkey bindings

Hotkey bindings may be read from external file (keymap-file). Initially, Midnight Commander creates key bindings using keymap defined in the source code. Then, two files `/usr/local/share/mc/mc.keymap` and `/usr/local/etc/mc/mc.keymap` are loaded always, sequentially reassigned key bindings defined earlier. User-defined keymap-file is searched on the following algorithm (to the first one found):

- 1) command line option **-K <keymap>** or **--keymap=<keymap>**
- 2) Environment variable **MC_KEYMAP**

- 3) Parameter **keymap** in section **[Midnight-Commander]** of config file.
- 4) File **~/.config/mc/mc.keymap**

Command line option, environment variable and parameter in config file may contain the absolute path to the keymap-file (with the extension .keymap or without it). Search of keymap-file will occur in (to the first one found):

- 1) **~/.config/mc**
- 2) **/usr/local/etc/mc/**
- 3) **/usr/local/share/mc/**

Miscellaneous Keys

Here are some keys which don't fall into any of the other categories:

- Enter** if there is some text in the command line (the one at the bottom of the panels), then that command is executed. If there is no text in the command line then if the selection bar is over a directory the Midnight Commander does a **chdir(2)** to the selected directory and reloads the information on the panel; if the selection is an executable file then it is executed. Finally, if the extension of the selected file name matches one of the extensions in the extensions file then the corresponding command is executed.
- C-l** repaint all the information in Midnight Commander.
- C-x c** run the Chmod command on a file or on the tagged files.
- C-x o** run the Chown command on the current file or on the tagged files.
- C-x l** run the hard link command.
- C-x s** run the absolute symbolic link command.
- C-x v** run the relative symbolic link command. See the File Menu section for more information about symbolic links.
- C-x i** set the other panel display mode to information.
- C-x q** set the other panel display mode to quick view.
- C-x !** execute the External panelize command.
- C-x h** run the add directory to hotlist command.
- Alt-!** executes the Filtered view command, described in the view command.
- Alt-?** executes the Find file command.
- Alt-c** pops up the quick cd dialog.
- C-o** when the program is being run in the Linux or FreeBSD console or under an xterm, it will show you the output of the previous command. When ran on the Linux console, Midnight Commander uses an external program (cons.saver) to handle saving and restoring of information on the screen.

When the subshell support is compiled in, you can type C-o at any time and you will be taken back to Midnight Commander's main screen, to return to your application just type C-o. If you have an application suspended by using this trick, you won't be able to execute other programs from Midnight Commander until you terminate the suspended application.

Directory Panels

This section lists the keys which operate on the directory panels. If you want to know how to change the appearance of the panels take a look at the section on Left and Right Menus.

Tab, C-i

change the current panel. The old other panel becomes the new current panel and the old current panel becomes the new other panel. The selection bar moves from the old current panel to the new current panel.

Insert, C-t

to tag files you may use the Insert key (the `kich1` terminfo sequence). To untag files, just retag a tagged file.

Alt-e to change charset of panel you may use Alt-e (M-e). Recoding is made from selected codepage into system codepage. To cancel the recoding, select "No translation" in the dialog of encodings.

Alt-g, Alt-r, Alt-j

used to select the top file in a panel, the middle file and the bottom one, respectively.

Alt-t toggle the current display listing to show the next display listing format. With this it is possible to quickly switch to brief listing, long listing, user defined listing format, and back to the default.

C-\ (control-backslash)

show the directory hotlist and change to the selected directory.

+ (plus)

this is used to select (tag) a group of files. Midnight Commander will prompt for a selection options. When *Files only* checkbox is on, only files will be selected. If *Files only* is off, as files as directories will be selected. When *Shell Patterns* checkbox is on, the regular expression is much like the filename globbing in the shell (* standing for zero or more characters and ? standing for one character). If *Shell Patterns* is off, then the tagging of files is done with normal regular expressions (see `ed (1)`). When *Case sensitive* checkbox is on, the selection will be case sensitive characters. If *Case sensitive* is off, the case will be ignored.

\ (backslash)

use the "\ " key to unselect a group of files. This is the opposite of the Plus key.

up-key, C-p

move the selection bar to the previous entry in the panel.

down-key, C-n

move the selection bar to the next entry in the panel.

home, a1, Alt-<

move the selection bar to the first entry in the panel.

end, c1, Alt->

move the selection bar to the last entry in the panel.

next-page, C-v

move the selection bar one page down.

prev-page, Alt-v

move the selection bar one page up.

Alt-o If the currently selected file is a directory, load that directory on the other panel and moves the selection to the next file. If the currently selected file is not a directory, load the parent directory on the other panel and moves the selection to the next file.

Alt-i make the current directory of the current panel also the current directory of the other panel. Put the other panel to the listing mode if needed. If the current panel is panelized, the other panel doesn't become panelized.

C-PageUp, C-PageDown

only when supported by the terminal: change to ".." and to the currently selected directory respectively.

Alt-y moves to the previous directory in the history, equivalent to clicking the < with the mouse.

Alt-u moves to the next directory in the history, equivalent to clicking the > with the mouse.

Alt-S-h, Alt-H

displays the directory history, equivalent to depressing the 'v' with the mouse.

Quick search

The Quick search mode allows you to perform fast file search in file panel. Press *C-s* or *Alt-s* to start a filename search in the directory listing.

When the search is active, the user input will be added to the search string instead of the command line. If the *Show mini-status* option is enabled the search string is shown on the mini-status line. When typing, the selection bar will move to the next file starting with the typed letters. The *Backspace* or *DEL* keys can be used to correct typing mistakes. If *C-s* is pressed again, the next match is searched for.

If quick search is started with double pressing of *C-s*, the previous quick search pattern will be used for current search.

Besides the filename characters, you can also use wildcard characters '*' and '?'.

Shell Command Line

This section lists keys which are useful to avoid excessive typing when entering shell commands.

Alt-Enter

copy the currently selected file name to the command line.

C-Enter

same as Alt-Enter. May not work on remote systems and some terminals.

C-S-Enter

copy the full path name of the currently selected file to the command line. May not work on remote systems and some terminals.

Alt-Tab

does the filename, command, variable, username and hostname completion for you.

C-x t, C-x C-t

copy the tagged files (or if there are no tagged files, the selected file) of the current panel (*C-x t*) or of the other panel (*C-x C-t*) to the command line.

C-x p, C-x C-p

the first key sequence copies the current path name to the command line, and the second one copies the unselected panel's path name to the command line.

C-q

the quote command can be used to insert characters that are otherwise interpreted by Midnight Commander (like the '+' symbol)

Alt-p, Alt-n

use these keys to browse through the command history. Alt-p takes you to the last entry, Alt-n takes you to the next one.

Alt-h displays the history for the current input line.

General Movement Keys

The help viewer, the file viewer and the directory tree use common code to handle moving. Therefore they accept exactly the same keys. Each of them also accepts some keys of its own.

Other parts of Midnight Commander use some of the same movement keys, so this section may be of use for those parts too.

Up, C-p

moves one line backward.

Down, C-n

moves one line forward.

Prev Page, Page Up, Alt-v

moves one page up.

Next Page, Page Down, C-v

moves one page down.

Home, A1

moves to the beginning.

End, C1

move to the end.

The help viewer and the file viewer accept the following keys in addition the to ones mentioned above:

b, C-b, C-h, Backspace, Delete

moves one page up.

Space bar

moves one page down.

u, d moves one half of a page up or down.

g, G moves to the beginning or to the end.

Input Line Keys

The input lines (they are used for the command line and for the query dialogs in the program) accept these keys:

C-a puts the cursor at the beginning of line.

C-e puts the cursor at the end of the line.

C-b, move-left

move the cursor one position left.

C-f, move-right

move the cursor one position right.

Alt-f moves one word forward.

Alt-b moves one word backward.

C-h, Backspace

delete the previous character.

C-d, Delete

delete the character in the point (over the cursor).

C-@ sets the mark for cutting.

C-w copies the text between the cursor and the mark to a kill buffer and removes the text from the input line.

Alt-w copies the text between the cursor and the mark to a kill buffer.

C-y yanks back the contents of the kill buffer.

C-k kills the text from the cursor to the end of the line.

Alt-p, Alt-n

Use these keys to browse through the command history. Alt-p takes you to the last entry, Alt-n takes you to the next one.

Alt-C-h, Alt-Backspace

delete one word backward.

Alt-Tab

does the filename, command, variable, username and hostname completion for you.

Menu Bar

The menu bar pops up when you press F9 or click the mouse on the top row of the screen. The menu bar has five menus: "Left", "File", "Command", "Options" and "Right".

The Left and Right Menus allow you to modify the appearance of the left and right directory panels.

The File Menu lists the actions you can perform on the currently selected file or the tagged files.

The Command Menu lists the actions which are more general and bear no relation to the currently selected file or the tagged files.

The Options Menu lists the actions which allow you to customize Midnight Commander.

Left and Right (Above and Below) Menus

The outlook of the directory panels can be changed from the **Left** and **Right** menus (they are named **Above** and **Below** when the horizontal panel split is chosen from the Layout options dialog).

Listing Format...

The listing mode view is used to display a listing of files, there are four different listing formats available: **Full**, **Brief**, **Long** and **User**. The full directory view shows the file name, the size of the file and the modification time.

The brief view shows only the file name and it has from 1 up to 9 columns (therefore showing more files unlike other views). The long view is similar to the output of **ls -l** command. The long view takes the whole screen width.

If you choose the "User" display format, then you have to specify the display format.

The user display format must start with a panel size specifier. This may be "half" or "full", and they specify a half screen panel and a full screen panel respectively.

After the panel size, you may specify how many listings to fit in the panel, side-by-side (in other words: how many times to repeat the fields horizontally). This defaults to 1. You may change this by adding a number from 1 to 9 to the format string.

After this you add the name of the fields with an optional size specifier. This are the available fields you may display:

- name** displays the file name.
- size** displays the file size.
- bsize** is an alternative form of the **size** format. It displays the size of the files and for directories it just shows SUB-DIR or UP--DIR.
- type** displays a one character wide type field. This character is similar to what is displayed by **ls** with the **-F** flag - * for executable files, / for directories, @ for links, = for sockets, - for character devices, + for block devices, | for pipes, ~ for symbolic links to directories and ! for stale symlinks (links that point nowhere).
- mark** an asterisk if the file is tagged, a space if it's not.
- mtime** file's last modification time.
- atime** file's last access time.
- ctime** file's status change time.
- perm** a string representing the current permission bits of the file.
- mode** an octal value with the current permission bits of the file.
- nlink** the number of links to the file.
- ngid** the GID (numeric).
- nuid** the UID (numeric).
- owner** the owner of the file.
- group** the group of the file.
- inode** the inode of the file.

Also you can use following keywords to define the panel layout:

space a space in the display format.

| add a vertical line to the display format.

To force one field to a fixed size (a size specifier), you just add **:** followed by the number of characters you want the field to have. If the number is followed by the symbol **+**, then the size specifies the minimal field size – if the program finds out that there is more space on the screen, it will then expand that field.

For example, the **Full** display corresponds to this format:

```
half type name | size | mtime
```

And the **Long** display corresponds to this format:

```
full perm space nlink space owner space group space size space mtime space name
```

This is a nice user display format:

```
half name | size:7 | type mode:3
```

Panels may also be set to the following modes:

Info The info view display information related to the currently selected file and if possible information about the current file system.

Tree The tree view is quite similar to the directory tree feature. See the section about it for more information.

Quick View

In this mode, the panel will switch to a reduced viewer that displays the contents of the currently selected file, if you select the panel (with the tab key or the mouse), you will have access to the usual viewer commands.

Sort Order...

The eight sort orders are by name, by extension, by modification time, by access time, and by inode information modification time, by size, by inode and unsorted. In the Sort order dialog box you can choose the sort order and you may also specify if you want to sort in reverse order by checking the reverse box.

By default directories are sorted before files but this can be changed from the Panel options menu (option **Mix all files**).

Filter...

The filter command allows you to specify a shell pattern (for example ***.tar.gz**) which the files must match to be shown. Regardless of the filter pattern, the directories and the links to directories are always shown in the directory panel.

Reread

The reread command reload the list of files in the directory. It is useful if other processes have created or removed files.

File Menu

Midnight Commander uses the F1 – F10 keys as keyboard shortcuts for commands appearing in the file menu. The escape sequences for the function keys are terminfo capabilities **kf1** through **kf10**. On terminals without function key support, you can achieve the same functionality by pressing the Esc key and then a number in the range 1 through 9 and 0 (corresponding to F1 to F9 and F10 respectively).

The File menu has the following commands (keyboard shortcuts in parentheses):

Help (F1)

Invokes the built-in hypertext help viewer. Inside the help viewer, you can use the Tab key to select the next link and the Enter key to follow that link. The keys Space and Backspace are used to move forward and backward in a help page. Press F1 again to get the full list of accepted keys.

Menu (F2)

Invoke the user menu. The user menu provides an easy way to provide users with a menu and add extra features to Midnight Commander.

View (F3, F13)

View the currently selected file. By default this invokes the Internal File Viewer but if the option "Use internal view" is off, it invokes an external file viewer specified by the **VIEWER** environment variable. If **VIEWER** is undefined, the **PAGER** environment variable is tried. If **PAGER** is also undefined, the "view" command is invoked. If you use F13 instead, the viewer will be invoked without doing any formatting or preprocessing to the file.

See parameters for external viewer for explain how you may specify an extended command line options for external viewers.

Filtered View (Alt-!)

This command prompts for a command and its arguments (the argument defaults to the currently selected file name), the output from such command is shown in the internal file viewer.

Edit (F4, F14)

Press F4 to edit the highlighted file. Press F14 (usually F14) to start the editor with a new, empty file. Currently they invoke the **vi** editor, or the editor specified in the **EDITOR** environment variable, or the Internal File Editor if the `use_internal_edit` option is on.

See parameters for external editor for explain how you may specify an extended command line options for external editors.

Copy (F5, F15)

Press F5 to pop up an input dialog to copy the currently selected file (or the tagged files, if there is at least one file tagged) to the directory/filename you specify in the input dialog. The destination defaults to the directory in the non-selected panel. Space for destination file may be preallocated relative to `preallocate_space` configure option. During this process, you can press C-c or Esc to abort the operation. For details about source mask (which will be usually either * or `^(.*)$` depending on setting of Use shell patterns) and possible wildcards in the destination see `Mask copy/rename`.

F15 (usually F15) is similar, but defaults to the directory in the selected panel. It always operates on the selected file, regardless of any tagged files.

On some systems, it is possible to do the copy in the background by clicking on the background button (or pressing Alt-b in the dialog box). The Background Jobs is used to control the background process.

Link (C-x l)

Create a hard link to the current file.

Absolute symlink (C-x s)

Create a absolute symbolic link to the current file.

Relative symLink (C-x v)

Create a relative symbolic link to the current file.

To those of you who don't know what links are: creating a link to a file is a bit like copying the file, but both the source filename and the destination filename represent the same file image. For example, if you edit one of these files, all changes you make will appear in both files. Some people call links aliases or shortcuts.

A hard link appears as a real file. After making it, there is no way of telling which one is the original and which is the link. If you delete either one of them the other one is still intact. It is very difficult to notice that the files represent the same image. Use hard links when you don't even want to know.

A symbolic link is a reference to the name of the original file. If the original file is deleted the symbolic link is useless. It is quite easy to notice that the files represent the same image. Midnight Commander shows an "@"-sign in front of the file name if it is a symbolic link to somewhere (except to directory, where it shows a tilde (~)). The original file which the link points to is shown on mini-status line if the `Show mini-status` option is enabled. Use symbolic links when you want to avoid the confusion that can be caused by hard

links.

When you press "C-x s" Midnight Commander will automatically fill in the complete path+filename of the original file and suggest a name for the link. You can change either one.

Sometimes you may want to change the absolute path of the original into a relative path. An absolute path starts from the root directory:

```
/home/frodo/mc/mc -> /home/frodo/new/mc
```

A relative link describes the original file's location starting from the location of the link itself:

```
/home/frodo/mc/mc -> ../new/mc
```

You can force Midnight Commander to suggest a relative path by pressing "C-x v" instead of "C-x s".

Rename/Move (F6, F16)

Press F6 to pop up an input dialog to copy the currently selected file (or the tagged files, if there is at least one file tagged) to the directory/filename you specify in the input dialog. The destination defaults to the directory in the non-selected panel. For more details look at Copy (F5) operation above, most of the things are quite similar.

F16 (usually F16) is similar, but defaults to the directory in the selected panel. It always operates on the selected file, regardless of any tagged files.

On some systems, it is possible to do the copy in the background by clicking on the background button (or pressing Alt-b in the dialog box). The Background Jobs is used to control the background process.

Mkdir (F7)

Pop up an input dialog and creates the directory specified.

Delete (F8)

Delete the currently selected file or the tagged files in the currently selected panel. During the process, you can press C-c or Esc to abort the operation.

Quick cd (Alt-c) Use the quick cd command if you have full command line and want to cd somewhere.

Select group (+)

This is used to select (tag) a group of files. Midnight Commander will prompt for a selection options. When *Files only* checkbox is on, only files will be selected. If *Files only* is off, as files as directories will be selected. When *Shell Patterns* checkbox is on, the regular expression is much like the filename globbing in the shell (* standing for zero or more characters and ? standing for one character). If *Shell Patterns* is off, then the tagging of files is done with normal regular expressions (see ed (1)). When *Case sensitive* checkbox is on, the selection will be case sensitive characters. If *Case sensitive* is off, the case will be ignored.

Unselect group (\)

Used to unselect a group of files. This is the opposite of the *Select group* command.

Quit (F10, S-F10)

Terminate Midnight Commander. S-F10 is used when you want to quit and you are using the shell wrapper. S-F10 will not take you to the last directory you visited with Midnight Commander, instead it will stay at the directory where you started Midnight Commander.

Quick cd

This command is useful if you have a full command line and want to cd somewhere without having to yank and paste the command line. This command pops up a small dialog, where you enter everything you would enter after **cd** on the command line and then you press enter. This features all the things that are already in the internal cd command.

Command Menu

The Directory tree command shows a tree figure of the directories.

The "Find file" command allows you to search for a specific file.

The "Swap panels" command swaps the contents of the two directory panels.

The "Switch panels on/off" command shows the output of the last shell command. This works only on xterm and on Linux and FreeBSD console.

The "Compare directories" command compares the directory panels with each other. You can then use the Copy (F5) command to make the panels identical. There are three compare methods. The quick method compares only file size and file date. The thorough method makes a full byte-by-byte compare. The thorough method is not available if the machine does not support the mmap(2) system call. The size-only compare method just compares the file sizes and does not check the contents or the date times, it just checks the file size.

The "External panelize" allows you to execute an external program, and make the output of that program the contents of the current panel.

The "Command history" command shows a list of typed commands. The selected command is copied to the command line. The command history can also be accessed by typing Alt-p or Alt-n.

The "Directory hotlist" command makes changing of the current directory to often used directories faster.

The "Screen list" command shows a dialog window with the list of currently running internal editors, viewers and other MC modules that support this mode.

The "Edit extension file" command allows you to specify programs to executed when you try to execute, view, edit and do a bunch of other thing on files with certain extensions (filename endings).

The "Edit Menu File" command may be used for editing the user menu (which appears by pressing F2).

Directory Tree

The Directory Tree command shows a tree figure of the directories. You can select a directory from the figure and Midnight Commander will change to that directory.

There are two ways to invoke the tree. The real directory tree command is available from Commands menu. The other way is to select tree view from the Left or Right menu.

To get rid of long delays, Midnight Commander creates the tree figure by scanning only a small subset of all the directories. If the directory which you want to see is missing, move to its parent directory and press C-r (or F2).

You can use the following keys:

General movement keys
are accepted.

Enter. In the directory tree, exits the directory tree and changes to this directory in the current panel. In the tree view, changes to this directory in the other panel and stays in tree view mode in the current panel.

C-r, F2 (Rescan).

Rescan this directory. Use this when the tree figure is out of date: it is missing subdirectories or shows some subdirectories which don't exist any more.

F3 (Forget).

Delete this directory from the tree figure. Use this to remove clutter from the figure. If you want the directory back to the tree figure press F2 in its parent directory.

F4 (Static/Dynamic).

Toggle between the dynamic navigation mode (default) and the static navigation mode.

In the static navigation mode you can use the Up and Down keys to select a directory. All known directories are shown.

In the dynamic navigation mode you can use the Up and Down keys to select a sibling directory, the Left key to move to the parent directory, and the Right key to move to a child directory. Only the parent, sibling and children directories are shown, others are left out. The tree figure changes dynamically as you traverse.

F5 (Copy).

Copy the directory.

F6 (RenMov).

Move the directory.

F7 (Mkdir).

Make a new directory below this directory.

F8 (Delete).

Delete this directory from the file system.

C-s, Alt-s.

Search the next directory matching the search string. If there is no such directory these keys will move one line down.

C-h, Backspace.

Delete the last character of the search string.

Any other character.

Add the character to the search string and move to the next directory which starts with these characters. In the tree view you must first activate the search mode by pressing C-s. The search string is shown in the mini status line.

The following actions are available only in the directory tree. They aren't supported in the tree view.

F1 (Help).

Invoke the help viewer and show this section.

Esc, F10.

Exit the directory tree. Do not change the directory.

The mouse is supported. A double-click behaves like Enter. See also the section on mouse support.

Find File

The Find File feature first asks for the start directory for the search and the filename to be searched for. By pressing the Tree button you can select the start directory from the directory tree figure.

The "File name" input field contains a filename pattern to be searched for. It is interpreted as a shell pattern or as a regular expression depending on the state of the "Using shell patterns" checkbox. An empty value is valid and matches any file name.

The "Content" input field contains a string to search for within the files. Leave this field empty to disable searching file contents.

Option "Whole words" allows select only those files containing matches that form whole words. Like `grep -w`.

You can start the search by pressing the OK button. During the search you can stop from the Stop button and continue from the Start button.

You can browse the filelist with the up and down arrow keys. The Chdir button will change to the directory of the currently selected file. The Again button will ask for the parameters for a new search. The Quit button quits the search operation. The Panelize button will place the found files to the current directory panel so that you can do additional operations on them (view, copy, move, delete and so on). To return to the normal file listing, change directory to "..".

The 'Enable ignore directories' checkbox and input field below it allow one to set up the list of directories that should be skip during the search files (for example, you may want to avoid searches on a CD-ROM or on a NFS directory that is mounted across a slow link). List components must be separated with a colon, here is an example:

```
/cdrom:/nfs/wuarchive:/afs
```

Relative paths are supported also. The following example shows how to skip special directories of version control systems:

```
/cdrom:/nfs/wuarchive:/afs:.svn:.git:CVS
```

Attention: input field can contain a dot (.), this means the current absolute path.

You may consider using the External panelize command for some operations. Find file command is for simple queries only, while using External panelize you can do as mysterious searches as you would like.

External panelize

The External panelize allows you to execute an external program, and make the output of that program the contents of the current panel.

For example, if you want to manipulate in one of the panels all the symbolic links in the current directory, you can use external panelization to run the following command:

```
find . -type l -print
```

Upon command completion, the directory contents of the panel will no longer be the directory listing of the current directory, but all the files that are symbolic links.

If you want to panelize all of the files that have been downloaded from your FTP server, you can use this awk command to extract the file name from the transfer log files:

```
awk '$9 ~!/incoming/ { print $9 }' < /var/log/xferlog
```

You may want to save often used panelize commands under a descriptive name, so that you can recall them quickly. You do this by typing the command on the input line and pressing Add new button. Then you enter a name under which you want the command to be saved. Next time, you just choose that command from the list and do not have to type it again.

Hotlist

The Directory hotlist command shows the labels of the directories in the directory hotlist. Midnight Commander will change to the directory corresponding to the selected label. From the hotlist dialog, you can remove already created label/directory pairs and add new ones. To add new directories quickly, you can use the Add to hotlist command (C-x h), which adds the current directory into the directory hotlist, asking just for the label for the directory.

This makes cd to often used directories faster. You may consider using the CDPATH variable as described in internal cd command description.

Edit Extension File

This will invoke your editor on the file `~/config/mc/mc.ext`. The format of this file following:

All lines starting with # or empty lines are thrown away.

Lines starting in the first column should have following format:

keyword/expr, i.e. everything after the slash until new line is *expr*.

keyword can be:

shell – *expr* is an extension (no wildcards). File matches if its name ends with *expr*. Example: *shell/tar* matches **.tar*.

regex – *expr* is a regular expression. File matches if its name matches the regular expression.

directory

– *expr* is a regular expression. File matches if it is a directory and its name matches the regular expression.

type – *expr* is a regular expression. File matches if the output of *file %f* without the initial "filename:" part matches regular expression *expr*.

default – matches any file. *expr* is ignored.

include – denotes a common section. *expr* is the name of the section.

Other lines should start with a space or tab and should be of the format: *keyword=command* (with no spaces around =), where *keyword* should be: *Open* (invoked on Enter or double click), *View* (F3), *Edit* (F4) or

Include (to add rules from the common section). *command* is any one-line shell command, with the simple macro substitution.

Rules are matched from top to bottom, thus the order is important. If the appropriate action is missing, search continues as if this rule didn't match (i.e. if a file matches the first and second entry and View action is missing in the first one, then on pressing F3 the View action from the second entry will be used). *default* should match all the actions.

Background Jobs

This lets you control the state of any background Midnight Commander process (only copy and move files operations can be done in the background). You can stop, restart and kill a background job from here.

Edit Menu File

The user menu is a menu of useful actions that can be customized by the user. When you access the user menu, the file `.mc.menu` from the current directory is used if it exists, but only if it is owned by user or root and is not world-writable. If no such file found, `~/config/mc/menu` is tried in the same way, and otherwise mc uses the default system-wide menu `/usr/local/share/mc/mc.menu`.

The format of the menu file is very simple. Lines that start with anything but space or tab are considered entries for the menu (in order to be able to use it like a hot key, the first character should be a letter). All the lines that start with a space or a tab are the commands that will be executed when the entry is selected.

When an option is selected all the command lines of the option are copied to a temporary file in the temporary directory (usually `/usr/tmp`) and then that file is executed. This allows the user to put normal shell constructs in the menus. Also simple macro substitution takes place before executing the menu code. For more information, see macro substitution.

Here is a sample `mc.menu` file:

```
A      Dump the currently selected file
        od -c %f

B      Edit a bug report and send it to root
        I='mktemp ${MC_TMPDIR:-/tmp}/mail.XXXXXX' || exit 1
        vi $I
        mail -s "Midnight Commander bug" root < $I
        rm -f $I

M      Read mail
        emacs -f rmail

N      Read Usenet news
        emacs -f gnus

H      Call the info hypertext browser
        info

J      Copy current directory to other panel recursively
        tar cf - . | (cd %D && tar xvpf -)

K      Make a release of the current subdirectory
        echo -n "Name of distribution file: "
        read tar
        ln -s %d 'dirname %d'/$tar
        cd ..
        tar cvhf ${tar}.tar $tar

= f *.tar.gz | f *.tgz & t n
X      Extract the contents of a compressed tar file
```

```
tar xzvf %f
```

Default Conditions

Each menu entry may be preceded by a condition. The condition must start from the first column with a '=' character. If the condition is true, the menu entry will be the default entry.

Condition syntax: = <sub-cond>
 or: = <sub-cond> | <sub-cond> ...
 or: = <sub-cond> & <sub-cond> ...

Sub-condition is one of following:

| | |
|--------------|--|
| y <pattern> | syntax of current file matching pattern? (for edit menu only) |
| f <pattern> | current file matching pattern? |
| F <pattern> | other file matching pattern? |
| d <pattern> | current directory matching pattern? |
| D <pattern> | other directory matching pattern? |
| t <type> | current file of type? |
| T <type> | other file of type? |
| x <filename> | is it executable filename? |
| ! <sub-cond> | negate the result of sub-condition |

Pattern is a normal shell pattern or a regular expression, according to the shell patterns option. You can override the global value of the shell patterns option by writing "shell_patterns=x" on the first line of the menu file (where "x" is either 0 or 1).

Type is one or more of the following characters:

| | |
|---|------------------|
| n | not a directory |
| r | regular file |
| d | directory |
| l | link |
| c | character device |
| b | block device |
| f | FIFO (pipe) |
| s | socket |
| x | executable file |
| t | tagged |

For example 'rlf' means either regular file, link or fifo. The 't' type is a little special because it acts on the panel instead of the file. The condition '=t t' is true if there are tagged files in the current panel and false if not.

If the condition starts with '=?' instead of '=' a debug trace will be shown whenever the value of the condition is calculated.

The conditions are calculated from left to right. This means

```
= f *.tar.gz | f *.tgz & t n
```

is calculated as

```
( (f *.tar.gz) | (f *.tgz) ) & (t n)
```

Here is a sample of the use of conditions:

```
= f *.tar.gz | f *.tgz & t n
```

```
L      List the contents of a compressed tar-archive
      gzip -cd %f | tar xvf -
```

Addition Conditions

If the condition begins with '+' (or '+?') instead of '=' (or '=?') it is an addition condition. If the condition

is true the menu entry will be included in the menu. If the condition is false the menu entry will not be included in the menu.

You can combine default and addition conditions by starting condition with '+=' or '=+' (or '+=?' or '=+?' if you want debug trace). If you want to use two different conditions, one for adding and another for defaulting, you can precede a menu entry with two condition lines, one starting with '+' and another starting with '='.

Comments are started with '#'. The additional comment lines must start with '#', space or tab.

Options Menu

Midnight Commander has some options that may be toggled on and off in several dialogs which are accessible from this menu. Options are enabled if they have an asterisk or "x" in front of them.

The Configuration command pops up a dialog from which you can change most of settings of Midnight Commander.

The Layout command pops up a dialog from which you specify a bunch of options how mc looks like on the screen.

The Panel options command pops up a dialog from which you specify options of file manager panels.

The Confirmation command pops up a dialog from which you specify which actions you want to confirm.

The Appearance command pops up a dialog from which you specify the skin.

The Display bits command pops up a dialog from which you may select which characters is your terminal able to display.

The Learn keys command pops up a dialog from which you test some keys which are not working on some terminals and you may fix them.

The Virtual FS command pops up a dialog from which you specify some VFS related options.

The Save setup command saves the current settings of the Left, Right and Options menus. A small number of other settings is saved, too.

Configuration

The options in this dialog are divided into several groups: "File operation options", "Esc key mode", "Pause after run" and "Other options".

File operation options

Verbose operation. This toggles whether the file Copy, Rename and Delete operations are verbose (i.e., display a dialog box for each operation). If you have a slow terminal, you may wish to disable the verbose operation. It is automatically turned off if the speed of your terminal is less than 9600 bps.

Compute totals. If this option is enabled, Midnight Commander computes total byte sizes and total number of files prior to any Copy, Rename and Delete operations. This will provide you with a more accurate progress bar at the expense of some speed. This option has no effect, if *Verbose operation* is disabled.

Classic progressbar. If this option is enabled, the progressbar of Copy/Move/Delete operations is always grown from left to right. If disabled, the growing direction of progressbar follows to direction of Copy/Move/Delete operation: from left panel to right one and vice versa. Enabled by default.

Mkdir autoname. When you press F7 to create a new directory, the input line in popup dialog will be filled by name of current file or directory in active panel. Disabled by default.

Preallocate space. Preallocate space for whole target file, if possible, before copy operation. Disabled by default.

Esc key mode.

By default, Midnight Commander treats the Esc key as a key prefix. Therefore, you should press Esc code twice to exit a dialog. But there is a possibility to use a single press of Esc key for that action.

Single press. By default this option is disabled. If you'll enable it, the Esc key will act as a prefix key for set up time interval (see *Timeout* option below), and if no extra keys have arrived, then the Esc key is

interpreted as a cancel key (Esc Esc).

Timeout. This options is used to setup the time interval (in microseconds) for single press of Esc key. By default, this interval is one second (1000000 microseconds). Also the timeout can be set via **KEYBOARD_KEY_TIMEOUT_US** environment variable (also in microseconds), which has higher priority than Timeout option value.

Pause after run

After executing your commands, Midnight Commander can pause, so that you can examine the output of the command. There are three possible settings for this variable:

Never. Means that you do not want to see the output of your command. If you are using the Linux or FreeBSD console or an xterm, you will be able to see the output of the command by typing C-o.

On dumb terminals. You will get the pause message on terminals that are not capable of showing the output of the last command executed (any terminal that is not an xterm or the Linux console).

Always. The program will pause after executing all of your commands.

Other options

Use internal editor. If this option is enabled, the built-in file editor is used to edit files. If the option is disabled, the editor specified in the **EDITOR** environment variable is used. If no editor is specified, **vi** is used. See the section on the internal file editor.

Use internal viewer. If this option is enabled, the built-in file viewer is used to view files. If the option is disabled, the pager specified in the **PAGER** environment variable is used. If no pager is specified, the **view** command is used. See the section on the internal file viewer.

Ask new file name. If this option is enabled, file name is asked before open new file in editor.

Auto menus. If this option is enabled, the user menu will be invoked at startup. Useful for building menus for non-unixers.

Drop down menus. When this option is enabled, the pull down menus will be activated as soon as you press the F9 key. Otherwise, you will only get the menu title, and you will have to activate the menu either with the arrow keys or with the hotkeys. It is recommended if you are using hotkeys.

Shell Patterns. By default the Select, Unselect and Filter commands will use shell-like regular expressions. The following conversions are performed to achieve this: the '*' is replaced by '.*' (zero or more characters); the '?' is replaced by '.' (exactly one character) and '.' by the literal dot. If the option is disabled, then the regular expressions are the ones described in `ed(1)`.

Complete: show all. By default, Midnight Commander pops up all possible completions if the completion is ambiguous only when you press **Alt-Tab** for the second time. For the first time, it just completes as much as possible and beeps in the case of ambiguity. Enable this option if you want to see all possible completions even after pressing **Alt-Tab** the first time.

Rotating dash. If this option is enabled, the Midnight Commander shows a rotating dash in the upper right corner as a work in progress indicator.

Cd follows links. This option, if set, causes Midnight Commander to follow the logical chain of directories when changing current directory either in the panels, or using the `cd` command. This is the default behavior of `bash`. When unset, Midnight Commander follows the real directory structure, so `cd ..` if you've entered that directory through a link will move you to the current directory's real parent and not to the directory where the link was present.

Safe delete. If this option is enabled, deleting files and directory hotlist entries unintentionally becomes more difficult. The default selection in the confirmation dialogs for deletion changes from **Yes** to **No**. This option is disabled by default.

Safe overwrite. If this option is enabled, overwriting files unintentionally becomes more difficult. The default selection in the overwrite confirmation dialog changes from **Yes** to **No**. This option is disabled by default.

Auto save setup. If this option is enabled, when you exit Midnight Commander, the configurable options of Midnight Commander are saved in the `~/config/mc/ini` file.

Layout

The layout dialog gives you a possibility to change the general layout of screen. The options in this dialog are divided into several groups: "Panel split", "Console output" and "Other options".

Panel split

The rest of the screen area is used for the two directory panels. You can specify whether the area is split to the panels in *Vertical* or *Horizontal* direction. Panel layout can be changed using `Alt-`, (`Alt-comma`) short-cut.

Equal split. By default, panels have equal sizes. Using this option you can specify an unequal split.

Console output

On the Linux or FreeBSD console you can specify how many lines are shown in the output window. This option is available if Midnight Commander runs on native console only.

Other options

Menu bar visible. If enabled, main menu of Midnight Commander is always visible on the top row of screen above panels. Enabled by default.

Command prompt. If enabled, command line is available. Enabled by default.

Keybar visible. If enabled, 10 labels associated with F1–F10 keys are located at the bottom row of screen. Enabled by default.

Hintbar visible. If enabled, the one-line hints are visible below panels. Enabled by default.

XTerm window title. When run in a terminal emulator for X11, Midnight Commander sets the terminal window title to the current working directory and updates it when necessary. If your terminal emulator is broken and you see some incorrect output on startup and directory change, turn off this option. Enabled by default.

Show free space. If enabled, free space and total space of current file system is shown at the bottom frame of panel. Enabled by default.

Panel options

Main panel options

Show mini-status. If enabled, one line of status information about the currently selected item is shown at the bottom of the panels. Enabled by default.

Use SI size units. If this option is enabled, Midnight Commander will use SI prefixes (base 10) when displaying any byte sizes. If disabled (default), Midnight Commander will use IEC prefixes (base 2).

Mix all files. If this option is enabled, all files and directories are shown mixed together. If the option is disabled (default), directories (and links to directories) are shown at the beginning of the listing, and other files below.

Show backup files. If enabled, Midnight Commander will show files ending with a tilde. Otherwise, they won't be shown (like GNU's `ls` option `-B`). Enabled by default.

Show hidden files. If enabled, Midnight Commander will show all files that start with a dot (like `ls -a`). Disabled by default.

Fast directory reload. If this option is enabled, Midnight Commander will use a trick to determine if the directory contents have changed. The trick is to reload the directory only if the `i`-node of the directory has changed; this means that reloads only happen when files are created or deleted. If what changes is the `i`-node for a file in the directory (file size changes, mode or owner changes, etc) the display is not updated. In these cases, if you have the option on, you have to rescan the directory manually (with `C-r`). Disabled by default.

Mark moves down. If enabled, the selection bar will move down when you mark a file (with `Insert` key).

Enabled by default.

Reverse files only. Allow revert selection of files only. Enabled by default. If enabled, the reverse selection is applied to files only, not to directories. The selection of directories is untouched. If off, the reverse selection is applied to files as well to directories: all unselected items become selected, and vice versa.

Simple swap. If both panels contain file listing, simple swap means that panels exchange its screen positions: left panel become right one, and vice versa. If this option is unchecked, file listing panels exchange its content keeping listing format and sort options. Unchecked by default.

Auto save panels setup. If this option is enabled, when you exit Midnight Commander, the current settings of panels are saved in the `~/.config/mc/panels.ini` file. Disabled by default.

Navigation

Lynx-like motion. If this option is enabled, you may use the arrows keys to automatically `chdir` if the current selection is a subdirectory and the shell command line is empty. By default, this setting is off.

Page scrolling. If set (the default), panel will scroll by half the display when the cursor reaches the end or the beginning of the panel, otherwise it will just scroll a file at a time.

Center scrolling. If set, panel will scroll when the cursor reaches the middle of the panel column, only hitting the top or bottom of the panel when actually on the first or last file. This behavior applies when scrolling one file at a time, and does not apply to the page up/down keys.

Mouse page scrolling. Controls whenever scrolling with the mouse wheel is done by pages or line by line on the panels.

File highlight

You can specify whether *permissions* and *file types* should be highlighted with distinctive Colors. If the permission highlighting is enabled, the parts of the *perm* and *mode* display fields which apply to the user running Midnight Commander are highlighted with the color defined by the *selected* keyword. If the file type highlighting is enabled, file names are colored according to rules described in `/usr/local/etc/mc/file-highlight.ini` file. See `Filenames Highlight` for more info.

Quick search

You can specify how the Quick search mode should work: case insensitively, case sensitively or be matched to the panel sort order: case sensitive or not.

Confirmation

In this dialog you configure the confirmation options for file deletion, overwriting files, execution by pressing enter, quitting the program, directory hotlist entries deletion and history cleanup.

Appearance

In this dialog you can select the skin to be used and enable shadow for dialogs and drop down menus.

See the `Skins` section for technical details about the skin definition files.

Shadows. If this option is enabled, all dialogs and drop down menus will have a shadow.

Display bits

This is used to configure the range of visible characters on the screen. This setting may be 7-bits if your terminal/curses supports only seven output bits, ISO-8859-1 displays all the characters in the ISO-8859-1 map and full 8 bits is for those terminals that can display full 8 bit characters.

Learn keys

This dialog allows you to test and redefine functional keys, cursor arrows and some other keys to make them work properly on your terminal. They often don't, since many terminal databases are incomplete or broken.

You can move around with the Tab key and with the vi moving keys ('h' left, 'j' down, 'k' up and 'l' right). Once you press any cursor movement key and it is recognized, you can use that key as well.

You can test keys just by pressing each of them. When you press a key and it is recognized properly, OK

should appear next to the name of that key. Once a key is marked OK it starts working as usually, e.g. F1 pressed the first time will just check that the F1 key works, but after that it will show help. The same applies to the arrow keys. The Tab key should be working always.

If some keys do not work properly then you won't see OK appear after pressing one of these. Then you may want to redefine it. Do it by pressing the button with the name of that key (either by the mouse or by Enter or Space after selecting the button with Tab or arrows). Then a message box will appear asking you to press that key. Do it and wait until the message box disappears. If you want to abort, just press Escape once and wait.

When you finish with all the keys, you can Save them. The definitions for the keys you have redefined will be written into the [terminal:TERM] section of your `~/.config/mc/ini` file (where TERM is the name of your current terminal). The definitions of the keys that were already working properly are not saved.

Virtual FS

This option gives you control over the settings of the Virtual File System.

Midnight Commander keeps in memory the information related to some of the virtual file systems to speed up the access to the files in the file system (for example, directory listings fetched from FTP servers).

Also, in order to access the contents of compressed files (for example, compressed tar files), Midnight Commander needs to create temporary uncompressed files on your disk.

Since both the information in memory and the temporary files on disk take up resources, you may want to tune the parameters of the cached information to decrease your resource usage or to maximize the speed of access to frequently used file systems.

Because of the format of the tar archives, the *Tar filesystem* needs to read the whole file just to load the file entries. Since most tar files are usually kept compressed (plain tar files are species in extinction), the tar file system has to uncompress the file on the disk in a temporary location and then access the uncompressed file as a regular tar file.

Now, since we all love to browse files and tar files all over the disk, it's common that you will leave a tar file and then re-enter it later. Since decompression is slow, Midnight Commander will cache the information in memory for a limited time. When the timeout expires, all the resources associated with the file system are released. The default timeout is set to one minute.

The FTP File System (ftpfs) allows you to browse directories on remote FTP servers. It has several options.

ftp anonymous password is the password used when you login as "anonymous". Some sites require a valid e-mail address. On the other hand, you probably don't want to give your real e-mail address to untrusted sites, especially if you are not using spam filtering.

ftpfs keeps the directory listing it fetches from a FTP server in a cache. The cache expire time is configurable with the *ftpfs directory cache timeout* option. A low value for this option may slow down every operation on the ftpfs because every operation would require sending a request to the FTP server.

You can define an FTP proxy host for doing FTP. Note that most modern firewalls are fully transparent at least for passive FTP (see below), so FTP proxies are considered obsolete.

If *Always use ftp proxy* is not set, you can use the exclamation sign to enable proxy for certain hosts. See FTP File System for examples.

If this option is set, the program will do two things: consult the `/usr/local/lib/mc/mc.no_proxy` file for lines containing host names that are local (if the host name starts with a dot, it is assumed to be a domain) and to assume that any hostnames without dots in their names are directly accessible. All other hosts will be accessed through the specified FTP proxy.

You can enable using `~/.netrc` file, which keeps login names and passwords for ftp servers. See netrc (5) for the description of the `.netrc` format.

Use passive mode enables using FTP passive mode, when the connection for data transfer is initiated by the client, not by the server. This option is recommended and enabled by default. If this option is turned off,

the data connection is initiated by the server. This may not work with some firewalls.

Save Setup

At startup, Midnight Commander tries to load initialization information from the `~/.config/mc/ini` file. If this file doesn't exist, the system-wide file `/usr/local/etc/mc/mc.ini` is used. If this file doesn't exist, the system-wide file `/usr/local/share/mc/mc.ini` is used. If this file doesn't exist, MC uses the default settings.

The *Save Setup* command creates the `~/.config/mc/ini` file by saving the current settings of the Left, Right and Options menus.

If you activate the *auto save setup* option, MC will always save the current settings when exiting.

There also exist settings which can't be changed from the menus. To change these settings you have to edit the setup file with your favorite editor. See the section on Special Settings for more information.

Executing operating system commands

You may execute commands by typing them directly in Midnight Commander's input line, or by selecting the program you want to execute with the selection bar in one of the panels and hitting Enter.

If you press Enter over a file that is not executable, Midnight Commander checks the extension of the selected file against the extensions in the Extensions File. If a match is found then the code associated with that extension is executed. A very simple macro expansion takes place before executing the command.

The `cd` internal command

The `cd` command is interpreted by Midnight Commander, it is not passed to the command shell for execution. Thus it may not handle all of the nice macro expansion and substitution that your shell does, although it does some of them:

Tilde substitution. The (`~`) will be substituted with your home directory, if you append a username after the tilde, then it will be substituted with the login directory of the specified user.

For example, `~guest` is the home directory for the user `guest`, while `~/guest` is the directory `guest` in your home directory.

Previous directory. You can jump to the directory you were previously by using the special directory name `'-'` like this: `cd -`

CDPATH directories. If the directory specified to the `cd` command is not in the current directory, then Midnight Commander uses the value in the environment variable **CDPATH** to search for the directory in any of the named directories.

For example you could set your **CDPATH** variable to `~/src:/usr/src`, allowing you to change your directory to any of the directories inside the `~/src` and `/usr/src` directories, from any place in the file system by using its relative name (for example `cd linux` could take you to `/usr/src/linux`).

Macro Substitution

When accessing a user menu, or executing an extension dependent command, or running a command from the command line input, a simple macro substitution takes place.

The macros are:

`%i` The indent of blank space, equal the cursor column position. For edit menu only.

`%y` The syntax type of current file. For edit menu only.

`%k` The block file name.

`%e` The error file name.

`%m` The current menu name.

`%f` and `%p`

In file manager user menu: the current file name in selected panel. In `mcedit` user menu: the name of opened file.

| | |
|-------------------------------------|--|
| <code>%x</code> | The extension of current file name. |
| <code>%b</code> | The current file name without extension. |
| <code>%d</code> | The current directory name. |
| <code>%F</code> | The current file in the unselected panel. |
| <code>%D</code> | The directory name of the unselected panel. |
| <code>%t</code> | The currently tagged files. |
| <code>%T</code> | The tagged files in the unselected panel. |
| <code>%u</code> and <code>%U</code> | Similar to the <code>%t</code> and <code>%T</code> macros, but in addition the files are untagged. You can use this macro only once per menu file entry or extension file entry, because next time there will be no tagged files. |
| <code>%s</code> and <code>%S</code> | The selected files: The tagged files if there are any. Otherwise the current file. |
| <code>%cd</code> | This is a special macro that is used to change the current directory to the directory specified in front of it. This is used primarily as an interface to the Virtual File System. |
| <code>%view</code> | This macro is used to invoke the internal viewer. This macro can be used alone, or with arguments. If you pass any arguments to this macro, they should be enclosed in brackets. The arguments are: <i>ascii</i> to force the viewer into ascii mode; <i>hex</i> to force the viewer into hex mode; <i>nroff</i> to tell the viewer that it should interpret the bold and underline sequences of nroff; <i>unformatted</i> to tell the viewer to not interpret nroff commands for making the text bold or underlined. |
| <code>%%</code> | The <code>%</code> character |
| <code>%{some text}</code> | Prompt for the substitution. An input box is shown and the text inside the braces is used as a prompt. The macro is substituted by the text typed by the user. The user can press Esc or F10 to cancel. This macro doesn't work on the command line yet. |
| <code>%var{ENV:default}</code> | If environment variable <i>ENV</i> is unset, the <i>default</i> is substituted. Otherwise, the value of <i>ENV</i> is substituted. |

The subshell support

The subshell support is a compile time option, that works with the shells: bash, ash (BusyBox and Debian), tcsh, zsh and fish.

When the subshell support is active, Midnight Commander will spawn a concurrent copy of your shell (the one defined in the **SHELL** variable and if it is not defined, then the one in the `/etc/passwd` file) and run it in a pseudo terminal, instead of invoking a new shell each time you execute a command, the command will be passed to the subshell as if you had typed it. This also allows you to change the environment variables, use shell functions and define aliases that are valid until you quit Midnight Commander.

bash users may specify startup commands in `~/.local/share/mc/bashrc` (fallback `~/.bashrc`) and special keyboard maps in `~/.local/share/mc/inputrc` (fallback `~/.inputrc`).

ash/dash users (BusyBox or Debian) may specify startup commands in `~/.local/share/mc/ashrc` (fallback `~/.profile`).

tcsh, zsh, fish users cannot specify mc-specific startup commands at present. They have to rely on shell-specific startup files.

The following paragraphs are relevant only when the subshell support is active:

You can suspend applications at any time with the sequence `C-o` and jump back to Midnight Commander, if you interrupt an application, you will not be able to run other external commands until you quit the

application you interrupted.

The basic prompt displayed by Midnight Commander is of the form "user@host:current_path\$ ". When using a capable shell, like Bash, the prompt displayed by Midnight Commander will be the same prompt that you are currently using in your shell.

(There's a known problem when using fish: the prompt is displayed only in full screen mode (Ctrl-o), not when the panels are visible.)

The OPTIONS section has more information on how you can control subshell usage (-U/-u). Furthermore, to set a specific subshell different from your current SHELL variable or login shell defined in /etc/passwd, you may call MC like this: **SHELL=/bin/myshell mc**

Chmod

The Chmod window is used to change the attribute bits in a group of files and directories. It can be invoked with the C-x c key combination.

The Chmod window has two parts – *Permissions* and *File*.

In the File section are displayed the name of the file or directory and its permissions in octal form, as well as its owner and group.

In the Permissions section there is a set of check buttons which correspond to the file attribute bits. As you change the attribute bits, you can see the octal value change in the File section.

To move between the widgets (buttons and check buttons) use the *arrow keys* or the *Tab* key. To change the state of the check buttons or to select a button use *Space*. You can also use the hotkeys on the buttons to quickly activate them. Hotkeys are shown as highlighted letters on the buttons.

To set the attribute bits, use the Enter key.

When working with a group of files or directories, you just click on the bits you want to set or clear. Once you have selected the bits you want to change, you select one of the action buttons (Set marked or Clear marked).

Finally, to set the attributes exactly to those specified, you can use the **[Set all]** button, which will act on all the tagged files.

[Marked all] set only marked attributes to all selected files

[Set marked] set marked bits in attributes of all selected files

[Clean marked] clear marked bits in attributes of all selected files

[Set] set the attributes of one file

[Cancel] cancel the Chmod command

Chown

The Chown command is used to change the owner/group of a file. The hot key for this command is C-x o.

Advanced Chown

The Advanced Chown command is the Chmod and Chown command combined into one window. You can change the permissions and owner/group of files at once.

Chattr

The Chattr window is used to change the attributes of a group of files and directories on a Linux file system. It can be invoked with the C-x e key combination.

Not all attributes are supported or utilized by all filesystems. List of available attribute flags is represented as a set of check buttons which correspond to the attribute flags (see **chattr(1)** for details). As you change the attribute flags, you can see the symbolic value change below file name.

To move between the widgets (buttons and check buttons) use the *arrow keys* or the *Tab* key. To change the state of the check buttons or to select a button use **Space**.

To set the attributes, use the Enter key.

When working with a group of files or directories, you just click on the flags you want to set or clear. Once you have selected the flags you want to change, you select one of the action buttons (Set marked or Clear marked).

Finally, to set the attributes exactly to those specified, you can use the **[Set all]** button, which will act on all the tagged files.

[Marked all] set only marked attributes to all selected files.

[Set marked] set marked flags in attributes of all selected files.

[Clean marked] clear marked flags in attributes of all selected files.

[Set] set the attributes of one file.

[Cancel] cancel the Chattr command.

File Operations

When you copy, move or delete files, Midnight Commander shows the file operations dialog. It shows the files currently being processed and uses up to three progress bars. The file bar indicates the percentage of the current file that has been processed so far. The count bar shows how many of the tagged files have been handled. The bytes bar indicates the percentage of the total size of the tagged files that has been handled. If the verbose option is off, the file and bytes bars are not shown.

There are two buttons at the bottom of the dialog. Pressing the Skip button will skip the rest of the current file. Pressing the Abort button will abort the whole operation, the rest of the files are skipped.

There are three other dialogs which you can run into during the file operations.

The error dialog informs about error conditions and has three choices. Normally you select either the Skip button to skip the file or the Abort button to abort the operation altogether. You can also select the Retry button if you fixed the problem from another terminal.

The replace dialog is shown when you attempt to copy or move a file on the top of an existing file. The dialog shows the dates and sizes of the both files. Press the Yes button to overwrite the file, the No button to skip the file, the All button to overwrite all the files, the None button to never overwrite and the Update button to overwrite if the source file is newer than the target file. You can abort the whole operation by pressing the Abort button.

The recursive delete dialog is shown when you try to delete a directory which is not empty. Press the Yes button to delete the directory recursively, the No button to skip the directory, the All button to delete all the directories and the None button to skip all the non-empty directories. You can abort the whole operation by pressing the Abort button. If you selected the Yes or All button you will be asked for a confirmation. Type "yes" only if you are really sure you want to do the recursive delete.

If you have tagged files and perform an operation on them only the files on which the operation succeeded are untagged. Failed and skipped files are left tagged.

Mask Copy/Rename

The copy/move operations let you translate the names of files in an easy way. To do it, you have to specify the correct source mask and usually in the trailing part of the destination specify some wildcards. All the files matching the source mask are copied/renamed according to the target mask. If there are tagged files, only the tagged files matching the source mask are renamed.

There are other options which you can set:

Follow links

determines whether make the symlinks and hardlinks in the source directory (recursively in subdirectories) new links in the target directory or whether would you like to copy their content.

Dive into subdirs

determines the behavior when the source directory is about to be copied, but the target directory already exists. The default action is to copy the contents of the source directory into the target directory. Enabling this option causes copying the source directory itself into the target directory.

For example, you want to copy directory */foo* containing file *bar* to */bla/foo*, which is an already existing directory. Normally (when **Dive into subdirs** is not set), mc would copy file */foo/bar* into the file */bla/foo/bar*. By enabling this option the */bla/foo/foo* directory will be created, and */foo/bar* will be copied into */bla/foo/foo/bar*.

Preserve attributes

determines whether to preserve the permissions, timestamps and (if you are root) the ownership of the original files. If this option is not set, the current value of the umask will be respected.

Use shell patterns

When this option is on you can use the '*' and '?' wildcards in the source mask. They work like they do in the shell. In the target mask only the '*' and '<digit>' wildcards are allowed. The first '*' wildcard in the target mask corresponds to the first wildcard group in the source mask, the second '*' corresponds to the second group and so on. The '\1' wildcard corresponds to the first wildcard group in the source mask, the '\2' wildcard corresponds to the second group and so on all the way up to '\9'. The '\0' wildcard is the whole filename of the source file.

Two examples:

If the source mask is "*.tar.gz", the destination is "/bla/*.tgz" and the file to be copied is "foo.tar.gz", the copy will be "foo.tgz" in "bla".

Suppose you want to swap basename and extension so that "file.c" would become "c.file" and so on. The source mask for this is "/*.*" and the destination is "\2.\1".

Use shell patterns off

When the shell patterns option is off the MC doesn't do automatic grouping anymore. You must use '\(...\)' expressions in the source mask to specify meaning for the wildcards in the target mask. This is more flexible but also requires more typing. Otherwise target masks are similar to the situation when the shell patterns option is on.

Two examples:

If the source mask is "\(.*)\tar\gz\$", the destination is "/bla/*.tgz" and the file to be copied is "foo.tar.gz", the copy will be "bla/foo.tgz".

Let's suppose you want to swap basename and extension so that "file.c" will become "c.file" and so on. The source mask for this is "\(.*)\(.*)\$" and the destination is "\2.\1".

Case Conversions

You can also change the case of the filenames. If you use 'u' or 'l' in the target mask, the next character will be converted to uppercase or lowercase correspondingly.

If you use 'U' or 'L' in the target mask, the next characters will be converted to uppercase or lowercase correspondingly up to the next 'E' or next 'U', 'L' or the end of the file name.

The 'u' and 'l' are stronger than 'U' and 'L'.

For example, if the source mask is '*' (*Use shell patterns* on) or '\(.*)\$' (*Use shell patterns* off) and the target mask is '\Lu*' the file names will be converted to have initial upper case and otherwise lower case.

You can also use '\' as a quote character. For example, '\\' is a backslash and '*' is an asterisk.

Stable symlinks

commands Midnight Commander, that it should change symlinks in the target, so that they'll point to the same location as it did before. With absolute symbolic links this does nothing, but if you have a relative one, it will recompute its value, adding necessary ../ and other directory parts and making the value as short as possible (most modern filesystems keep short symlinks inside inodes and thus don't waste much disk space).

Select/Unselect Files

The dialog of group of files and directories selection or unselection. The input line allow enter the regular expression of filenames that will be selected/unselected.

When *Files only* checkbox is on, only files will be selected. If *Files only* is off, as files as directories will be selected. When *Shell Patterns* checkbox is on, the regular expression is much like the filename globbing in the shell (* standing for zero or more characters and ? standing for one character). If *Shell Patterns* is off, then the tagging of files is done with normal regular expressions (see `ed (1)`). When *Case sensitive* checkbox is on, the selection will be case sensitive characters. If *Case sensitive* is off, the case will be ignored.

Internal Diff Viewer

The `mcdiff` is a visual diff tool. You can compare two files and edit them in-place (diffs are updated dynamically). You can browse and view a working copy from popular version control systems (GIT, Subversion, etc).

Following shortcuts are available in internal diff viewer of Midnight Commander.

- F1** Invoke the built-in hypertext help viewer.
- F2** Save modified files.
- F4** Edit file of the left panel in the internal editor.
- F14** Edit file of the right panel in the internal editor.
- F5** Merge the current hunk. Only the current hunk will be merged.
- F7** Start search.
- F17** Continue search.
- F10, Esc, q**
Exit from diff viewer.
- Alt-s, s**
Toggle show of hunk status.
- Alt-n, l**
Toggle show of line numbers.
- f** Maximize left panel.
- =** Make panels equal in width.
- >** Reduce the size of the right panel.
- <** Reduce the size of the left panel.
- c** Toggle show of trailing carriage return (CR) symbol as ^M.
- 2, 3, 4, 8**
Set tabulation size
- C-u** Swap contents of diff panels.
- C-r** Refresh the screen.
- C-o** Switch to the subshell and show the command screen.
- Enter, Space, n**
Find next diff hunk.
- Backspace, p**
Find previous diff hunk.
- g** Go to line.
- Down** Scroll one line forward.

Up Scroll one line backward.

PageUp
Move one page up.

PageDown
Moves one page down.

Home, A1
Moves to the line beginning.

End Moves to the line end.

C-Home
Move to the file beginning.

C-End, C1
Move to the file end.

Internal File Viewer

The internal file viewer provides two display modes: ASCII and hex. To toggle between modes, use the F4 key.

The viewer will try to use the best method provided by your system or the file type to display the information. Some character sequences, which appear most often in preformatted manual pages, are displayed bold and underlined, thus making a pretty display of your files.

When in hex mode, the search function accepts text in quotes and constant numbers. Text in quotes is matched exactly after removing the quotes. Each number matches one byte. You can mix quoted text with constants like this:

"String" 34 0xBB 012 "more text"

Numbers are always interpreted in hex. In the example above, "34" is interpreted as 0x34. The prefix "0x" isn't really needed: we could type "BB" instead of "0xBB". And "012" is interpreted as 0x12, not as an octal number.

Here is a listing of the actions associated with each key that the Midnight Commander handles in the internal file viewer.

F1 Invoke the built-in hypertext help viewer.

F2 Toggle the wrap mode.

F4 Toggle the hex mode.

F5 Goto. You can specify a line number, offset or percentage of file size of position that you want to view.

F7,/,? Start search. These keys call the dialog window that allows you to set up the search options. If key is ? the "Backwards" option is on.

C-s Continue forward search.

C-r Continue reverse search.

F17, n Continue search in the chosen direction.

N Temporary change the search direction: backwards if forward search is chosen, and vice versa.

F8 Toggle Raw/Parsed mode: This will show the file as found on disk or if a processing filter has been specified in the mc.ext file, then the output from the filter. Current mode is always the other than written on the button label, since on the button is the mode which you enter by that key.

F9 Toggle the format/unformat mode: when format mode is on the viewer will interpret some string sequences to show bold and underline with different colors. Also, on button label is the other mode than current.

F10, Esc.

Exit the internal file viewer.

PageDown, space, C-v.

Scroll one page forward.

PageUp, Alt-v, C-b, Backspace.

Scroll one page backward.

Down

Scroll one line forward.

Up

Scroll one line backward.

C-l

Refresh the screen.

C-o

Switch to the subshell and show the command screen.

[n] m

Set the mark n.

[n] r

Jump to the mark n.

C-f

Jump to the next file.

C-b

Jump to the previous file.

Alt-r

Toggle the ruler.

Alt-e

to change charset of displayed text may use Alt-e (M-e). Recoding is made from selected codepage into system codepage. To cancel the recoding you may select "<No translation>" in charset selection dialog.

It's possible to instruct the file viewer how to display a file, look at the Edit Extension File section

Internal File Editor

The internal file editor is a full-featured full screen editor. It can edit files up to 64 megabytes. It is possible to edit binary files. The internal file editor is invoked using **F4** if the `use_internal_edit` option is set in the initialization file.

The features it presently supports are: block copy, move, delete, cut, paste; key for key undo; pull-down menus; file insertion; macro commands; regular expression search and replace; S-arrow text highlighting (if supported by the terminal); insert-overwrite toggle; word wrap; autoindent; tunable tab size; syntax highlighting for various file types; and an option to pipe text blocks through shell commands like indent and ispell.

Sections:

Options of editor in ini-file

The editor is very easy to use and requires no tutoring. To see what keys do what, just consult the appropriate pull-down menu. Other keys are: Shift movement keys do text highlighting. **C-Ins** copies to the file **mcedit.clip** and **S-Ins** pastes from **mcedit.clip**. **S-Del** cuts to **mcedit.clip**, and **C-Del** deletes highlighted text. Mouse highlighting also works, and you can override the mouse as usual by holding down the shift key while dragging the mouse to let normal terminal mouse highlighting work.

To define a macro, press **C-R** and then type out the key strokes you want to be executed. Press **C-R** again when finished. You can then assign the macro to any key you like by pressing that key. The macro is executed when you press **C-A** and then the assigned key. The macro is also executed if you press Meta, Ctrl, or Esc and the assigned key, provided that the key is not used for any other function. Once defined, the macro commands go into the file `~/local/share/mc/mcedit/mcedit.macros`. You can delete a macro by deleting the appropriate line in this file.

To change charset of displayed text may use Alt-e (M-e). Recoding is made from selected codepage into system codepage. To cancel the recoding you may select "<No translation>" in charset selection dialog.

F19 will format the currently highlighted block (plain text or C or C++ code or another). This is controlled by the file `/usr/local/share/mc/edit.indent.rc` which is copied to `~/local/share/mc/mcedit/edit.indent.rc`

in your home directory the first time you use it.

The editor also displays non-us characters (160+). When editing binary files, you should set **display bits** to 7 bits in the options menu to keep the spacing clean.

Options of editor in ini-file

Some editor options of ini-file are described in this section. Options are placed in [Midnight-Commander] section

editor_wordcompletion_collect_entire_file

Search autocomplete candidates in entire of file or just from begin of file to cursor position (0)

Screen selector

Midnight Commander supports running many internal modules (such as editor, viewer and diff viewer) simultaneously and switching between them without closing open files. Using several file managers at a time, however, is not currently supported.

Let's call each of these modules a screen. There are three ways to switch between screens, using one of these global shortcuts:

Alt-} switch to the next screen;

Alt-{ switch to the previous screen;

Alt-' open a dialog window with the list of currently open screens (or use the "Screen list" menu item).

Completion

Let Midnight Commander type for you.

Attempt to perform completion on the text before current position. MC attempts completion treating the text as variable (if the text begins with \$), username (if the text begins with ~), hostname (if the text begins with @) or command (if you are on the command line in the position where you might type a command, possible completions then include shell reserved words and shell built-in commands as well) in turn. If none of these matches, filename completion is attempted.

Filename, username, variable and hostname completion works on all input lines, command completion is command line specific. If the completion is ambiguous (there are more different possibilities), MC beeps and the following action depends on the setting of the Complete: show all option in the Configuration dialog. If it is enabled, a list of all possibilities pops up next to the current position and you can select with the arrow keys and **Enter** the correct entry. You can also type the first letters in which the possibilities differ to move to a subset of all possibilities and complete as much as possible. If you press **Alt-Tab** again, only the subset will be shown in the listbox, otherwise the first item which matches all the previous characters will be highlighted. As soon as there is no ambiguity, dialog disappears, but you can hide it by canceling keys **Esc**, **F10** and left and right arrow keys. If Complete: show all is disabled, the dialog pops up only if you press **Alt-Tab** for the second time, for the first time MC just beeps.

Apply escaping of **?**, *****, and **&** symbols (as **\?**, *****, and **\&**) in filenames to disallow use them as metasymbols in regular expressions when substitution is performed in the input line.

Virtual File System

Midnight Commander is provided with a code layer to access the file system; this code layer is known as the virtual file system switch. The virtual file system switch allows Midnight Commander to manipulate files not located on the Unix file system.

Currently, Midnight Commander is packaged with some Virtual File Systems (VFS): the *local* file system, used for accessing the regular Unix file system; the *ftpfs*, used to manipulate files on remote systems with the FTP protocol; the *tarfs*, used to manipulate tar and compressed tar files; the *undelfs*, used to recover deleted files on ext2 file systems (the default file system for Linux systems), *fish* (for manipulating files over shell connections such as rsh and ssh). If the code was compiled with *sftpfs* (for manipulating files over SFTP connections). If the code was compiled with *smbfs* support, you can manipulate files on remote

systems with the SMB (CIFS) protocol.

A generic *extfs* (EXternal virtual File System) is provided in order to easily expand VFS capabilities using scripts and external software.

The VFS switch code will interpret all of the path names used and will forward them to the correct file system, the formats used for each one of the file systems is described later in their own section.

FTP File System

The FTP File System (ftpfs) allows you to manipulate files on remote machines. To actually use it, you can use the *FTP link* item in the menu or directly change your current directory using the *cd* command to a path name that looks like this:

```
ftp://[!][user[:pass]]@[machine[:port]][remote-dir]
```

The *user*, *port* and *remote-dir* elements are optional. If you specify the *user* element, Midnight Commander will login to the remote machine as that user, otherwise it will use anonymous login or the login name from the *~/.netrc* file. The optional *pass* element is the password used for the connection. Using the password in the VFS directory name is not recommended, because it can appear on the screen in clear text and can be saved to the directory history.

To enable using FTP proxy, prepend ! (an exclamation sign) to the hostname.

Examples:

```
ftp://ftp.nuclecu.unam.mx/linux/local
ftp://tsx-11.mit.edu/pub/linux/packages
ftp://!behind.firewall.edu/pub
ftp://guest@remote-host.com:40/pub
ftp://miguel:xxx@server/pub
```

Please check the Virtual File System dialog box for ftpfs options.

Tar File System

The tar file system provides you with read-only access to your tar files and compressed tar files by using the *chdir* command. To change your directory to a tar file, you change your current directory to the tar file by using the following syntax:

```
/filename.tar/utar://[dir-inside-tar]
```

The *mc.ext* file already provides a shortcut for tar files, this means that usually you just point to a tar file and press return to enter into the tar file, see the Edit Extension File section for details on how this is done.

Examples:

```
mc-3.0.tar.gz/utar://mc-3.0/vfs
/ftp/GCC/gcc-2.7.0.tar/utar://
```

The latter specifies the full path of the tar archive.

File transfer over SHell filesystem

The fish file system is a network based file system that allows you to manipulate the files in a remote machine as if they were local. To use this, the other side has to either run fish server, or has to have bash-compatible shell.

To connect to a remote machine, you just need to *chdir* into a special directory which name is in the following format:

```
sh://[user@]machine[:options]/[remote-dir]
```

The *user*, *options* and *remote-dir* elements are optional. If you specify the *user* element, Midnight Commander will try to login on the remote machine as that user, otherwise it will use your login name.

The available *options* are:

- 'C' – use compression;
- 'r' – use rsh instead of ssh;

`port` – specify the port used by remote server.
 If the `remote-dir` element is present, your current directory on the remote machine will be set to this one.

Examples:

```
sh://onlyrsh.mx:r/linux/local
sh://joe@want.compression.edu:C/private
sh://joe@noncompressed.ssh.edu/private
sh://joe@somehost.ssh.edu:2222/private
```

SFTP (SSH File Transfer Protocol) filesystem

The SFTP file system is a network based file system that allows you to manipulate the files in a remote machine as if they were local.

To connect to a remote machine, you just need to `chdir` into a special directory which name is in the following format:

```
sftp://[user@]machine:[port]/[remote-dir]
```

The `user`, `port` and `remote-dir` elements are optional. If you specify the `user` element, Midnight Commander will try to login on the remote machine as that user, otherwise it will use your login name. `port` – specify the port used by remote server (22 by default). If the `remote-dir` element is present, your current directory on the remote machine will be set to this one.

Examples:

```
sftp://onlyrsh.mx/linux/local
sftp://joe:password@want.compression.edu/private
sftp://joe@noncompressed.ssh.edu/private
sftp://joe@somehost.ssh.edu:2222/private
```

Undelete File System

On Linux systems, if you asked configure to use the ext2fs undelete facilities, you will have the undelete file system available. Recovery of deleted files is only available on ext2 file systems. The undelete file system is just an interface to the ext2fs library to retrieve all of the deleted files names on an ext2fs and provides and to extract the selected files into a regular partition.

To use this file system, you have to `chdir` into the special file name formed by the "undel://" prefix and the file name where the actual file system resides.

For example, to recover deleted files on the second partition of the first SCSI disk on Linux, you would use the following path name:

```
undel://sda2
```

It may take a while for the `undelfs` to load the required information before you start browsing files there.

SMB File System

The `smbfs` allows you to manipulate files on remote machines with SMB (or CIFS) protocol. These include Windows for Workgroups, Windows 9x/ME/XP, Windows NT, Windows 2000 and Samba. To actually use it, you may try to use the panel command "SMB link..." (accessible from the menubar) or you may directly change your current directory to it using the `cd` command to a path name that looks like this:

```
smb://[user@]machine[/service]/[remote-dir]
```

The `user`, `service` and `remote-dir` elements are optional. The `user`, `domain` and `password` can be specified in an input dialog.

Examples:

```
smb://machine/Share
smb://other_machine
smb://guest@machine/Public/Irlex
```

EXtErnal File System

extfs allows you to integrate numerous features and file types into GNU Midnight Commander in an easy way, by writing scripts.

Extfs filesystems can be divided into two categories:

1. Stand-alone filesystems, which are not associated with any existing file. They represent certain system-wide data as a directory tree. You can invoke them by typing `cd fsname://` where `fsname` is an extfs short name (see below). Examples of such filesystems include `audio` (list audio tracks on the CD) or `apt` (list of all Debian packages in the system).

For example, to list CD–Audio tracks on your CD–ROM drive, type

```
cd audio://
```

2. 'Archive' filesystems (like `rpm`, `patchfs` and more), which represent contents of a file as a directory tree. It can consist of 'real' files compressed in an archive (`urar`, `rpm`) or virtual files, like messages in a mailbox (`mailfs`) or parts of a patch (`patchfs`). To access such filesystems `fsname://` should be appended to the archive name. Note that the archive itself can be on another vfs.

For example, to list contents of a zip archive `documents.zip` type

```
cd documents.zip/uzip://
```

In many aspects, you could treat extfs like any other directory. For instance, you can add it to the hotlist or change to it from directory history. An important limitation is that you cannot invoke shell commands inside extfs, just like any other non-local VFS.

Common extfs scripts included with Midnight Commander are:

- a** access 'A:' DOS/Windows diskette (`cd a://`).
- apt** front end to Debian's APT package management system (`cd apt://`).
- audio** audio CD ripping and playing (`cd audio://` or `cd device/audio://`).
- bpp** package of Bad Penguin GNU/Linux distribution (`cd file.bpp/bpp://`).
- deb** package of Debian GNU/Linux distribution (`cd file.deb/deb://`).
- dpkg** Debian GNU/Linux installed packages (`cd deb://`).
- hp48** view and copy files to/from a HP48 calculator (`cd hp48://`).
- lsIR** browsing of lsIR listings as found on many FTPs (`cd filename/lsIR://`).
- mailfs** mbox-style mailbox files support (`cd mailbox/mailfs://`).
- patchfs**
extfs to handle unified and context diffs (`cd filename/patchfs://`).
- rpm** RPM package (`cd filename/rpm://`).
- rpms** RPM database management (`cd rpms://`).
- ulha, urar, uzip, uzoo, uar, uha**
archivers (`cd archive/xxxx://` where `xxxx` is one of: `ulha`, `urar`, `uzip`, `uzoo`, `uar`, `uha`).

You could bind file type/extension to specified extfs as described in the Edit Extension File section. Here is an example entry for Debian packages:

```
regex/.deb$
Open=%cd %p/deb://
```

Colors

Midnight Commander will try to detect if your terminal supports color using the terminal database and your terminal name. Sometimes it gets confused, so you may force color mode or disable color mode using the `-c` and `-b` flag respectively.

If the program is compiled with the S-Lang screen manager instead of ncurses, it will also check the

variable **COLORTERM**, if it is set, it has the same effect as the `-c` flag.

You may specify terminals that always force color mode by adding the *color_terminals* variable to the Colors section of the initialization file. This will prevent Midnight Commander from trying to detect if your terminal supports color. Example:

```
[Colors]
color_terminals=linux,xterm
color_terminals=terminal-name1,terminal-name2...
```

The program can be compiled with both ncurses and S-Lang, ncurses does not provide a way to force color mode: ncurses uses just the information in the terminal database.

Midnight Commander provides a way to change the default colors. Currently the colors are configured using the environment variable **MC_COLOR_TABLE** or the Colors section in the initialization file.

In the Colors section, the default color map is loaded from the *base_color* variable. You can specify an alternate color map for a terminal by using the terminal name as the key in this section. Example:

```
[Colors]
base_color=
xterm=menu=magenta:marked=,magenta:markselect=,red
```

The format for the color definition is:

```
<keyword>=<fgcolor>,<bgcolor>,<attributes>:<keyword>=...
```

The colors are optional, and the keywords are: normal, selected, disabled, marked, markselect, errors, input, inputmark, inputunchanged, commandlinemark, reverse, gauge, header, inputhistory, commandhistory. Button bar colors are: bbarhotkey, bbarbutton. Status bar color: statusbar. Menu colors are: menunormal, menusel, menuhot, menuhotsel, menuinactive. Dialog colors are: dnormal, dfocus, dhotnormal, dhotfocus, dttitle. Error dialog colors are: errdfocus, errdhotnormal, errdhotfocus, errdttitle. Help colors are: helpnormal, helpitalic, helpbold, helplink, helpslink, helptitle. Viewer colors are: viewnormal, viewbold, viewunderline, viewselected. Editor colors are: editnormal, editbold, editmarked, editwhitespace, editlinestate. Popup menu colors are: pmenunormal, pmenusel, pmenutitle.

header determines the color of panel header, the line that contains column titles and sort mode indicator.

input determines the color of input lines used in query dialogs.

gauge determines the color of the filled part of the progress bar (gauge), which is used to show the user the progress of file operations, such as copying.

disabled determines the color of the widget that cannot be selected.

The dialog boxes use the following colors: *dnormal* is used for the normal text, *dfocus* is the color used for the currently selected component, *dhotnormal* is the color used to differentiate the hotkey color in normal components, whereas the *dhotfocus* color is used for the highlighted color in the currently selected component.

Menus use the same scheme but uses the menunormal, menusel, menuhot, menuhotsel and menuinactive tags instead.

Help uses the following colors: *helpnormal* is used for normal text, *helpitalic* is used for text which is emphasized in italic in the manual page, *helpbold* is used for text which is emphasized in bold in the manual page, *helplink* is used for not selected hyperlinks and *helpslink* is used for selected hyperlink.

Popup menu uses following colors: *pmenunormal* is used for non-selected menu items and as a main color of popup menu window, *pmenusel* is used for selected menu item, *pmenutitle* is used for popup menu title.

The possible colors are: black, gray, red, brightred, green, brightgreen, brown, yellow, blue, brightblue, magenta, brightmagenta, cyan, brightcyan, lightgray and white. And there is a special keyword for transparent background. It is 'default'. The 'default' can only be used for background color. Another special keyword "base" means mc's main colors. When 256 colors are available, they can be specified either as color16 to color255, or as rgb000 to rgb555 and gray0 to gray23. Example:

[Colors]

base_color=normal=white,default:marked=magenta,default

Attributes can be any of bold, italic, underline, reverse and blink, appended by a plus sign if more than one are desired. The special word "none" means no attributes, without attempting to fall back to base_color.

Example:

menuhotsel=yellow:black:bold+underline

Skins

You can change the appearance of Midnight Commander. To do this, you must specify a file that contain descriptions of colors and lines to draw boxes. Redefining of the colors is entirely compatible with the assignment of colors, as described in Section Colors.

If your skin contains any true-color definitions, you should define the 'truecolors' key set to TRUE value in [skin] section. If true-color is not used but 256-color is, you should define '256colors' instead.

A skin-file is searched on the following algorithm (to the first one found):

- 1) command line option **-S <skin>** or **--skin=<skin>**
- 2) Environment variable **MC_SKIN**
- 3) Parameter **skin** in section **[Midnight-Commander]** in config file.
- 4) File **/usr/local/etc/mc/skins/default.ini**
- 5) File **/usr/local/share/mc/skins/default.ini**

Command line option, environment variable and parameter in config file may contain the absolute path to the skin-file (with the extension .ini or without it). Search of skin-file will occur in (to the first one found):

- 1) **~/local/share/mc/skins/**
- 2) **/usr/local/etc/mc/skins/**
- 3) **/usr/local/share/mc/skins/**

For getting extended info, refer to:

Description of section and parameters
Color pair definitions
Color and attribute aliases
Draw lines
Compatibility

Description of section and parameters

Section **[skin]** contain metainfo for skin-file. Parameter *description* contain short text about skin.

Section **[filehighlight]** contain descriptions of color pairs for filenames highlighting. Name of parameters must be equal to names of sections into filehighlight.ini file. See Filenames Highlight for getting more info.

Section **[core]** describes the elements that are used everywhere.

default

Default color pair. Used in all other sections if they not contain color definitions

selected

cursor

marked

selected data

markselect
 cursor on selected data

gauge color of the filled part of the progress bar

input color of input lines used in query dialogs

inputmark
 color of input selected text

inputunchanged
 color of input text before first modification or cursor movement

commandlinemark
 color of selected text in command line

reverse reverse color

Section **[dialog]** describes the elements that are placed on dialog windows (except error dialogs).

default
 Default color for this section. Used [core]._default_ if not specified

dfocus Color of active element (in focus)

dhotnormal
 Color of hotkeys

dhotfocus
 Color of hotkeys in focused element

Section **[error]** describes the elements that are placed on error dialog windows

default
 Default color for this section. Used [core]._default_ if not specified

errdhotnormal
 Color of hotkeys

errdhotfocus
 Color of hotkeys in focused element

Section **[menu]** describes the elements that are placed in menu. This section describes system menu (called by F9) and user-defined menus (called by F2 in panels and by F11 in editor).

default
 Default color for this section. Used [core]._default_ if not specified

entry Color of menu items

menuhot
 Color of menu hotkeys

menusel
 Color of active menu item (in focus)

menuhotsel
 Color of menu hotkeys in focused menu item

menuinactive
 Color of inactive menu

Section **[help]** describes the elements that are placed on help window.

default
Default color for this section. Used [core]._default_ if not specified

helpitalic
Color pair for element with **italic** attribute

helpbold
Color pair for element with **bold** attribute

helplink
Color of links

helpslink
Color of active link (on focus)

Section [**editor**] describes the colors of elements placed in editor.

default
Default color for this section. Used [core]._default_ if not specified

editbold
Color pair for element with **bold** attribute

editmarked
Color of selected text

editwhitespace
Color of tabs and trailing spaces highlighting

editlinestate
Color for line state area

Section [**viewer**] describes the colors of elements placed in viewer.

viewunderline
Color pair for element with **underline** attribute

Color pair definitions

Any parameter in skin-file contain definition of color pair.

Color pairs described as two colors and the optional attributes separated by ';'. First field sets the foreground color, second field sets background color, third field sets the attributes. Any of the fields may be omitted, in this case value will be taken from default color pair (global color pair or from default color pair of this section).

Example:

```
[core]
# green on black
_default_=green:black
# green (default) on blue
selected=;blue
# yellow on black (default)
# underlined yellow on black (default)
marked=yellow;;underline
```

Possible colors (names) and attributes are described in Colors. section.

Color and attribute aliases

This optional section might define aliases for single colors (not color pairs) as well as combination of attributes; in other words, for semicolon-separated fragments of parameters. Aliases can refer to other

aliases as long as they don't form a loop.

Example:

```
[aliases]
  myfavfg=green
  myfavbg=black
  myfavattr=bold+italic
[core]
  _default_=myfavfg;myfavbg;myfavattr
```

Draw lines

Lines sets in section **[Lines]** into skin-file. By default single lines are used, but you may redefine to usage of any utf-8 symbols (like to lines, for example).

WARNING!!! When you build Midnight Commander with the ncurses screen library usage of drawing lines is limited! Possible only drawing a single lines. For all questions and comments please contact the developers of ncurses.

Descriptions of parameters **[Lines]**:

lefttop left-top line fragment.

righttop right-top line fragment.

centertop down branch of horizontal line

centerbottom up branch of horizontal line

leftbottom left-bottom line fragment

rightbottom right-bottom line fragment

leftmiddle right branch of vertical line

rightmiddle left branch of vertical line

centermiddle cross of lines

horiz horizontal line

vert vertical line

thinhoriz thin horizontal line

thinvert thin vertical line

Compatibility

Appointment of color by skin-files fully compatible with the appointment of the colors described in Colors. section.

In this case, reassignment of colors has priority over the skin file and is complementary.

Filenames Highlight

Section [filehighlight] in current skin-file contains key names as highlight groups and values as color pairs. Color pairs is documented in Skins section.

Rules of filenames highlight are placed in /usr/local/share/mc/filehighlight.ini file (~/.config/mc/filehighlight.ini). Name of section in this file must be equal to parameters names in [filehighlight] section (in current skin-file).

Keys in these groups are:

type file type. If present, all other options are ignored.

regexp regular expression. If present, 'extensions' option is ignored.

extensions

list of extensions of files. Separated by ';' sign.

extensions_case

(make sense only with 'extensions' parameter) make 'extensions' rule case sensitive (true) or not (false).

'type' key may have values:

- FILE (all files)
- FILE_EXE
- DIR (all directories)
- LINK_DIR
- LINK (all links except stale link)
- HARDLINK
- SYMLINK
- STALE_LINK
- DEVICE (all device files)
- DEVICE_BLOCK
- DEVICE_CHAR
- SPECIAL (all special files)
- SPECIAL_SOCKET
- SPECIAL_FIFO
- SPECIAL_DOOR

Special Settings

Most of Midnight Commander settings can be changed from the menus. However, there are a small number of settings which can only be changed by editing the setup file.

These variables may be set in your ~/.config/mc/ini file:

clear_before_exec

By default, Midnight Commander clears the screen before executing a command. If you would prefer to see the output of the command at the bottom of the screen, edit your ~/.config/mc/ini file and change the value of the field clear_before_exec to 0.

confirm_view_dir

If you press F3 on a directory, normally MC enters that directory. If this flag is set to 1, then MC will ask for confirmation before changing the directory if you have files tagged.

ftpfs_retry_seconds

This value is the number of seconds Midnight Commander will wait before attempting to reconnect to an FTP server that has denied the login. If the value is zero, the login will no be retried.

max_dirt_limit

Specifies how many screen updates can be skipped at most in the internal file viewer. Normally this value is not significant, because the code automatically adjusts the number of updates to skip according to the rate of incoming keystrokes. However, on very slow machines or terminals with a fast keyboard auto repeat, a big value can make screen updates too jumpy.

It seems that setting `max_dirt_limit` to 10 causes the best behavior, and that is the default value.

mouse_move_pages_viewer

Controls if scrolling with the mouse is done by pages or line by line on the internal file viewer.

only_leading_plus_minus

Allow special treatment for '+', '-', '*' in the command line (select, unselect, reverse selection) only if the command line is empty. You don't need to quote those characters in the middle of the command line. On the other hand, you cannot use them to change selection when the command line is not empty.

alternate_plus_minus

If true, use '+', '-', '\` and '*' keys normally. For select/unselect, use 'Alt++', 'Alt--' and 'Alt-*'.

show_output_starts_shell

This variable only works if you are not using the subshell support. When you use the C-o key-stroke to go back to the user screen, if this one is set, you will get a fresh shell. Otherwise, pressing any key will bring you back to Midnight Commander.

timeformat_recent

Change the time format used to display dates less than 6 months from now. See `strftime` or `date` man page for the format specification. If this option is absent, default `timeformat` is used.

timeformat_old

Change the time format used to display dates older than 6 months from now or for dates in the future. See `strftime` or `date` man page for the format specification. If this option is absent, default `timeformat` is used.

torben_fj_mode

If this flag is set, then the home and end keys will work slightly different on the panels, instead of moving the selection to the first and last files in the panels, they will act as follows:

The home key will: Go up to the middle line, if below it; else go to the top line unless it is already on the top line, in this case it will go to the first file in the panel.

The end key has a similar behavior: Go down to the middle line, if over it; else go to the bottom line unless you already are at the bottom line, in such case it will move the selection to the last file name in the panel.

use_file_to_guess_type

If this variable is on (the default) it will spawn the `file` command to match the file types listed on the `mc.ext` file.

xtree_mode

If this variable is on (default is off) when you browse the file system on a Tree panel, it will automatically reload the other panel with the contents of the selected directory.

fish_directory_timeout

This variable holds the lifetime of a directory cache entry in seconds. The default value is 900 seconds.

clipboard_store

This variable contains path (with options) to the external clipboard utility like `xclip` to read text into X selection from file. For example:

```
clipboard_store=xclip -i
```

clipboard_paste

This variable contains path (with options) to the external clipboard utility like `xclip` to print the selection to standard out. For example:

```
clipboard_paste=xclip -o
```

autodetect_codeset

This option allows use the 'enca' command to autodetect codeset of text files in internal viewer and editor. List of valid values can be obtain by the 'enca --list languages | cut -d : -f1' command. Option must be located in the [Misc] section.

For example:

```
autodetect_codeset=russian
```

Parameters for external editor or viewer

Midnight Commander provides a way for specify an options for external editors and viewers. Midnight Commander tries to search the "[External editor or viewer parameters]" section in the system initialization file (the mc.lib file located in Midnight Commander's library directory) and then in the ~/.config/mc/ini file. The option name should be equal to the name (full pathname) of external editor or viewer. The option value can contain following variables:

%filename

The filename to edit/view.

%lineno

The start line in the opening file.

For example:

```
[External editor or viewer parameters]
```

```
vi=%filename +%lineno
```

```
joe=%filename +%lineno
```

```
more=%filename +%lineno
```

Start line is passed to the external editor/viewer only if it is called from the Find file results window.

If external editor/viewer is launched via F4/F3 keys, MC hopes that program (at least "joe", but probably others too) has an own feature that by default opens the file where it was last open. MC doesn't prevent external editor/viewer to save and restore position in opened files.

Terminal databases

Midnight Commander provides a way to fix your system terminal database without requiring root privileges. Midnight Commander searches in the system initialization file (the mc.lib file located in Midnight Commander's library directory) and in the ~/.config/mc/ini file for the section "terminal:your-terminal-name" and then for the section "terminal:general", each line of the section contains a key symbol that you want to define, followed by an equal sign and the definition for the key. You can use the special \e form to represent the escape character and the ^x to represent the control-x character.

The possible key symbols are:

f0 to f20 Function keys f0-f20

bs backspace

home home key

end end key

up up arrow key

down down arrow key

left left arrow key

right right arrow key

pgdn page down key

pgup page up key

insert the insert character

delete the delete character

complete to do completion

For example, to define the key insert to be the Escape + [+ O + p, you set this in the ini file:

```
insert=\e[Op
```


Also now you can use *extended learn keys*. For example:

```
ctrl-alt-right=\e[[1;6C
ctrl-alt-left=\e[[1;6D
```

This means that ctrl+alt+left sends a `\e[[1;6D` escape sequence and therefore Midnight Commander interprets `"\e[[1;6D"` as C-Alt-Left.

The *complete* key symbol represents the escape sequences used to invoke the completion process, this is invoked with Alt-tab, but you can define other keys to do the same work (on those keyboard with tons of nice and unused keys everywhere).

FILES

Full paths below may vary between installations. They are also affected by the **MC_DATADIR** environment variable. If it's set, its value is used instead of `/usr/local/share/mc` in the paths below.

`/usr/local/share/mc/help/mc.hlp`

The help file for the program.

`/usr/local/share/mc/mc.ext`

The default system-wide extensions file.

`~/.config/mc/mc.ext`

User's own extension, view configuration and edit configuration file. They override the contents of the system wide files if present.

`/usr/local/etc/mc/mc.ini`

`/usr/local/share/mc/mc.ini`

System-wide setup files for Midnight Commander, used only if the user doesn't have his own `~/.config/mc.ini` file. If `/usr/local/etc/mc/mc.ini` exists, `/usr/local/share/mc/mc.ini` isn't used.

`/usr/local/share/mc/mc.lib`

Global settings for Midnight Commander. Settings in this file affect all users, whether they have `~/.config/mc.ini` or not. Currently, only terminal settings are loaded from `mc.lib`.

`~/.config/mc/ini`

User's own setup. If this file is present then the setup is loaded from here instead of the system-wide startup file.

`/usr/local/share/mc/hints/mc.hint`

This file contains the hints displayed by the program.

`/usr/local/share/mc/mc.menu`

This file contains the default system-wide applications menu.

`~/.config/mc/menu`

User's own application menu. If this file is present it is used instead of the system-wide applications menu.

`~/.cache/mc/Tree`

The directory list for the directory tree and tree view features.

`~/.local/share/mc.menu`

Local user-defined menu. If this file is present, it is used instead of the home or system-wide applications menu.

To change default root directory of MC, you can use **MC_PROFILE_ROOT** environment variable. The value of **MC_PROFILE_ROOT** must be an absolute path. If **MC_PROFILE_ROOT** is unset or empty, **HOME** variable is used. If **HOME** is unset or empty, MC directories are get from GLib library.

LICENSE

This program is distributed under the terms of the GNU General Public License as published by the Free Software Foundation. See the built-in help for details on the License and the lack of warranty.

AVAILABILITY

The latest version of this program can be found at <http://ftp.midnight-commander.org/>.

SEE ALSO

`ed(1)`, `gpm(1)`, `terminfo(1)`, `view(1)`, `sh(1)`, `bash(1)`, `tcsh(1)`, `zsh(1)`.

Midnight Commander's page on the World Wide Web:

<http://www.midnight-commander.org/>

AUTHORS

Authors and contributors are listed in the **AUTHORS** file in the source distribution.

BUGS

See the file **TODO** in the distribution for information on what remains to be done.

If you want to report a problem with the program, please create bugreport at <http://www.midnight-commander.org/>.

Provide a detailed description of the bug, the version of the program you are running (*mc -V* displays this information), the operating system you are running the program on. If the program crashes, we would appreciate a stack trace.

NAME

`mcedit` – Internal file editor of GNU Midnight Commander.

SYNOPSIS

`mcedit` [`-bcDfhstVx?`] [`+lineno`] [`file1`] [`file2`] ...

`mcedit` [`-bcDfhstVx?`] `file1:lineno[:]` `file2:lineno[:]` ...

DESCRIPTION

`mcedit` is a link to **mc**, the main GNU Midnight Commander executable. Executing GNU Midnight Commander under this name runs the internal editor and opens files specified on the command line. The editor is based on the terminal version of **cooledit** – standalone editor for X Window System.

OPTIONS

`+lineno`

Go to the line specified by number (do not put a space between the + sign and the number). Several line numbers are allowed but only the last one will be used, and it will be applied to the first file only.

`-b` Force black and white display.

`-c` Force ANSI color mode on terminals that don't seem to have color support.

`-C` `<keyword>=<fgcolor>,<bgcolor>,<attributes>:<keyword>= ...`
Specify a different color set. See the **Colors** section in `mc(1)` for more information.

`-d` Disable mouse support.

`-f` Display the compiled-in search path for GNU Midnight Commander data files.

`-t` Force using termcap database instead of terminfo. This option is only applicable if GNU Midnight Commander was compiled with S-Lang library with terminfo support.

`-V` Display the version of the program.

`-x` Force xterm mode. Used when running on xterm-capable terminals (two screen modes, and able to send mouse escape sequences).

FEATURES

The internal file editor is a full-featured windowed editor. It can edit several files at the same time. Maximum size of each file is 64 megabytes. It is possible to edit binary files. The features it presently supports are: block copy, move, delete, cut, paste; key for key undo; pull-down menus; file insertion; macro commands; regular expression search and replace; shift-arrow text highlighting (if supported by the terminal); insert-overwrite toggle; autoindent; tunable tab size; syntax highlighting for various file types; and an option to pipe text blocks through shell commands like `indent` and `ispell`.

Each file is opened in its own window in full-screen mode. Window control in `mcedit` is similar to the window control in other multi-window program: double click on window title maximizes the window to full-screen or restores window size and position; left-click on window title and mouse drag moves the window in editor area; left-click on low-right frame corner and mouse drag resizes the window. These actions can be made using "Window" menu.

KEYS

The editor is easy to use and can be used without learning. The pull-down menu is invoked by pressing F9. You can learn other keys from the menu and from the button bar labels.

In addition to that, Shift combined with arrows does text highlighting (if supported by the terminal): **Ctrl-Ins** copies to the file `~/cache/mc/mcedit/mcedit.clip`, **Shift-Ins** pastes from `~/cache/mc/mcedit/mcedit.clip`, **Shift-Del** cuts to `~/cache/mc/mcedit/mcedit.clip`, and **Ctrl-Del** deletes highlighted text. Mouse highlighting also works on some terminals. To use the standard mouse support provided by your terminal, hold the Shift key. Please note that the mouse support in the terminal doesn't share the clipboard with **mcedit**.

The completion key (usually **Meta-Tab** or **Escape Tab**) completes the word under the cursor using the words used in the file.

MACRO

To define a macro, press **Ctrl-R** and then type out the keys you want to be executed. Press **Ctrl-R** again when finished. The macro can be assigned to any key by pressing that key. The macro is executed when you press the assigned key.

The macro commands are stored in section **[editor]** in the file `~/.local/share/mc/mc.macros`.

External scripts (filters) can be assigned into the any hotkey by edit **mc.macros** like following:

```
[editor]
ctrl-W=ExecuteScript:25;
```

This means that `ctrl-W` hotkey initiates the *ExecuteScript(25)* action, then editor handler translates this into execution of `~/.local/share/mc/mcedit/macros.d/macro.25.sh` shell script.

External scripts are stored in `~/.local/share/mc/mcedit/macros.d/` directory and must be named as **macro.XXXX.sh** where **XXXX** is the number from 0 to 9999. See **Edit Menu File** for more detail about format of the script.

Following macro definition and directives can be used:

#silent If this directive is set, then script starts without interactive subshell.

%c The cursor column position number.

%i The indent of blank space, equal the cursor column.

%y The syntax type of current file.

%b The block file name.

%f The current file name.

%n Only the current file name without extension.

%x The extension of current file name.

%d The current directory name.

%F The current file in the unselected panel.

%D The directory name of the unselected panel.

%t The currently tagged files.

%T The tagged files in the unselected panel.

%u and **%U**

Similar to the **%t** and **%T** macros, but in addition the files are untagged. You can use this macro only once per menu file entry or extension file entry, because next time there will be no tagged files.

%s and **%S**

The selected files: The tagged files if there are any. Otherwise the current file.

Feel free to edit this files, if you need. Here is a sample external script:

```
1 comment selection
   TMPFILE='mktemp ${MC_TMPDIR:-/tmp}/up.XXXXXX' || exit 1
   echo #if 0 > $TMPFILE
   cat %b >> $TMPFILE
   echo #endif >> $TMPFILE
   cat $TMPFILE > %b
   rm -f $TMPFILE
```

If some keys don't work, you can use **Learn Keys** in the **Options** menu.

CODE NAVIGATION

mcedit can be used for navigation through code with tags files created by `etags` or `ctags` commands. If there is no TAGS file code navigation will not work. For example, in case of `exuberant-ctags` for C language command will be:

```
ctags -e --language-force=C -R ./
```

Meta-Enter shows list box to select item under cursor (cursor should stand at the end of the word).

Meta-Minus where minus is symbol "-" goes to previous function in navigation list (like browser's Back button).

Meta-Equal where equal is symbol "=" goes to next function in navigation list (like browser's Forward button).

SYNTAX HIGHLIGHTING

mcedit supports syntax highlighting. This means that keywords and contexts (like C comments, string constants, etc) are highlighted in different colors. The following section explains the format of the file `~/.config/mc/mcedit/Syntax`. If this file is missing, system-wide `/usr/local/share/mc/syntax/Syntax` is used. The file `~/.config/mc/mcedit/Syntax` is rescanned on opening of every new editor file. The file contains rules for highlighting, each of which is given on a separate line, and define which keywords will be highlighted with what color.

The file is divided into sections, each beginning with a line with the **file** command. The sections are normally put into separate files using the **include** command.

The **file** command has three arguments. The first argument is a regular expression that is applied to the file name to determine if the following section applies to the file. The second argument is the description of the file type. It is used in **cooledit**; future versions of **mcedit** may use it as well. The third optional argument is a regular expression to match the first line of text of the file. The rules in the following section apply if either the file name or the first line of text matches.

A section ends with the start of another section. Each section is divided into contexts, and each context contains rules. A context is a scope within the text that a particular set of rules belongs to. For instance, the text within a C style comment (i.e. between `/*` and `*/`) has its own color. This is a context, although it has no further rules inside it because there is probably nothing that we want highlighted within a C comment.

A trivial C programming section might look like this:

```
file .*\\c C\\sProgram\\sFile (#include|/\\\\*)
```

```
wholechars abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_
```

```
# default colors
```

```
define comment brown
```

```
context default
```

```
keyword whole if yellow
keyword whole else yellow
keyword whole for yellow
keyword whole while yellow
keyword whole do yellow
keyword whole switch yellow
keyword whole case yellow
keyword whole static yellow
keyword whole extern yellow
keyword { brightcyan
keyword } brightcyan
keyword '*' green
```

```
# C comments
context /\* \*/ comment
```

```
# C preprocessor directives
context linestart # \n red
keyword \\n brightred
```

```
# C string constants
context " " green
keyword %d brightgreen
keyword %s brightgreen
keyword %c brightgreen
keyword \" brightgreen
```

Each context starts with a line of the form:

```
context [exclusive] [whole|wholeright|wholeleft] [linestart] delim [linestart] delim [foreground] [background] [attributes]
```

The first context is an exception. It must start with the command

```
context default [foreground] [background] [attributes]
```

otherwise **mcedit** will report an error. The **linestart** option specifies that *delim* must start at the beginning of a line. The **whole** option tells that *delim* must be a whole word. To specify that a word must begin on the word boundary only on the left side, you can use the **wholeleft** option, and similarly a word that must end on the word boundary is specified by **wholeright**.

The set of characters that constitute a whole word can be changed at any point in the file with the **wholechars** command. The left and right set of characters can be set separately with

```
wholechars [left|right] characters
```

The **exclusive** option causes the text between the delimiters to be highlighted, but not the delimiters themselves.

Each rule is a line of the form:

```
keyword [whole|wholeright|wholeleft] [linestart] string foreground [background] [attributes]
```

Context or keyword strings are interpreted, so that you can include tabs and spaces with the sequences `\t` and `\s`. Newlines and backslashes are specified with `\n` and `\\` respectively. Since whitespace is used as a separator, it may not be used as is. Also, `*` must be used to specify an asterisk. The `*` itself is a wildcard that matches any length of characters. For example,

```
keyword      '* '    green
```

colors all C single character constants green. You also could use

```
keyword      "*"    green
```

to color string constants, but the matched string would not be allowed to span across multiple newlines. The wildcard may be used within context delimiters as well, but you cannot have a wildcard as the last or first character.

Important to note is the line

```
keyword \\n brightgreen
```

This line defines a keyword containing the backslash and newline characters. Since the keywords are matched before the context delimiters, this keyword prevents the context from ending at the end of the lines that end in a backslash, thus allowing C preprocessor directive to continue across multiple lines.

The possible colors are: black, gray, red, brightred, green, brightgreen, brown, yellow, blue, brightblue, magenta, brightmagenta, cyan, brightcyan, lightgray and white. The special keyword "default" means the terminal's default. Another special keyword "base" means mc's main colors, it is useful as a placeholder if

you want to specify attributes without modifying the background color. When 256 colors are available, they can be specified either as color16 to color255, or as rgb000 to rgb555 and gray0 to gray23.

If the syntax file is shared with **cooledit**, it is possible to specify different colors for **mcedit** and **cooledit** by separating them with a slash, e.g.

```
keyword #include red/Orange
```

mcedit uses the color before the slash. See cooledit(1) for supported **cooledit** colors.

Attributes can be any of bold, italic, underline, reverse and blink, appended by a plus sign if more than one are desired.

Comments may be put on a separate line starting with the hash sign (#).

If you are describing case insensitive language you need to use **caseinsensitive** directive. It should be specified at the beginning of syntax file.

Because of the simplicity of the implementation, there are a few intricacies that will not be dealt with correctly but these are a minor irritation. On the whole, a broad spectrum of quite complicated situations are handled with these simple rules. It is a good idea to take a look at the syntax file to see some of the nifty tricks you can do with a little imagination. If you cannot get by with the rules I have coded, and you think you have a rule that would be useful, please email me with your request. However, do not ask for regular expression support, because this is flatly impossible.

A useful hint is to work with as much as possible with the things you can do rather than try to do things that this implementation cannot deal with. Also remember that the aim of syntax highlighting is to make programming less prone to error, not to make code look pretty.

The syntax highlighting can be toggled using Ctrl-s shortcut.

COLORS

The default colors may be changed by appending to the **MC_COLOR_TABLE** environment variable. Foreground and background colors pairs may be specified for example with:

```
MC_COLOR_TABLE="$MC_COLOR_TABLE:\
editnormal=lightgray,black:\
editbold=yellow,black:\
editmarked=black,cyan"
```

OPTIONS

Most options can be set from Options dialog box. See the **Options** menu. The following options are defined in **./config/mc/ini** and have obvious counterparts in the dialog box. You can modify them to change the editor behavior, by editing the file. Unless specified, a 1 sets the option to on, and a 0 sets it to off, as usual.

use_internal_edit

This option is ignored when invoking **mcedit**.

editor_tab_spacing

Interpret the tab character as being of this length. Default is 8. You should avoid using other than 8 since most other editors and text viewers assume a tab spacing of 8. Use **editor_fake_half_tabs** to simulate a smaller tab spacing.

editor_fill_tabs_with_spaces

Never insert a tab character. Rather insert spaces (ascii 32) to fill to the desired tab size.

editor_return_does_auto_indent

Pressing return will tab across to match the indentation of the first line above that has text on it.

editor_backspace_through_tabs

Make a single backspace delete all the space to the left margin if there is no text between the cursor and the left margin.

editor_fake_half_tabs

This will emulate a half tab for those who want to program with a tab spacing of 4, but do not want the tab size changed from 8 (so that the code will be formatted the same when displayed by other programs). When editing between text and the left margin, moving and tabbing will be as though a tab space were 4, while actually using spaces and normal tabs for an optimal fill. When editing anywhere else, a normal tab is inserted.

editor_option_save_mode

Possible values 0, 1 and 2. The save mode (see the options menu also) allows you to change the method of saving a file. Quick save (0) saves the file immediately, truncating the disk file to zero length (i.e. erasing it) and then writing the editor contents to the file. This method is fast, but dangerous, since a system error during a file save will leave the file only partially written, possibly rendering the data irretrievable. When saving, the safe save (1) option enables creation of a temporary file into which the file contents are first written. In the event of a problem, the original file is untouched. When the temporary file is successfully written, it is renamed to the name of the original file, thus replacing it. The safest method is create backups (2): a backup file is created before any changes are made. You can specify your own backup file extension in the dialog. Note that saving twice will replace your backup as well as your original file.

editor_word_wrap_line_length

Line length to wrap at. Default is 72.

editor_backup_extension

Symbol to add to name of backup files. Default is "~".

editor_line_state

Show state line of editor. Currently it shows current line number (in the future it might show things like folding, breakpoints, etc.). M-n toggles this option.

editor_visible_spaces

Toggle "show visible trailing spaces". If editor_visible_spaces=1, they are shown as ' '.

editor_visible_tabs

Toggle "show visible tabs". If editor_visible_tabs=1, tabs are shown as '<---->'.

editor_persistent_selections

Do not remove block selection after cursor movement.

editor_drop_selection_on_copy

Reset selection after copy to clipboard.

editor_cursor_beyond_eol

Allow moving cursor beyond the end of line.

editor_cursor_after_inserted_block

Allow moving cursor after inserted block.

editor_syntax_highlighting

enable syntax highlighting.

editor_edit_confirm_save

Show confirmation dialog on save.

editor_option_ttypewriter_wrap

to be described

editor_option_auto_para_formatting

to be described

editor_option_save_position

Save file position on exit.

source_codepage

Symbol representation of codepage name for file (i.e. CP1251, ~ – default).

editor_group_undo

Combine UNDO actions for several of the same type of action (inserting/overwriting, deleting, navigating, typing)

editor_wordcompletion_collect_entire_file

Search autocomplete candidates in entire file (1) or just from beginning of file to cursor position (0).

spell_language

Spelling language (en, en-variant_0, ru, etc) installed with aspell package (a full list can be obtained using 'aspell' utility). Use **spell_language = NONE** to disable aspell support. Default value is 'en'. Option must be located in the [Misc] section.

editor_stop_format_chars

Set of characters to stop paragraph formatting. If one of those characters is found in the beginning of line, that line and all following lines of paragraph will be untouched. Default value is "-+*,\.,:;>".

editor_state_full_filename

Show full path name in the status line. If disabled (default), only base name of the file is shown.

MISCELLANEOUS

The editor also displays non-us characters (160+). When editing binary files, you should set **display bits** to 7 bits in Midnight Commander's options menu to keep the spacing clean.

FILES

/usr/local/share/mc/help/mc.hlp

The help file for the program.

/usr/local/share/mc/mc.ini

The default system-wide setup for GNU Midnight Commander, used only if the user's own *~/config/mc/ini* file is missing.

/usr/local/share/mc/mc.lib

Global settings for Midnight Commander. Settings in this file affect all users, whether they have *~/config/mc/ini* or not.

*/usr/local/share/mc/syntax/**

The default system-wide syntax files for mcedit, used only if the corresponding user's own *~/local/share/mc/mcedit/* file is missing.

~/config/mc/ini

User's own setup. If this file is present then the setup is loaded from here instead of the system-wide setup file.

~/local/share/mc/mcedit/

User's own directory where block commands are processed and saved and user's own syntax files are located.

LICENSE

This program is distributed under the terms of the GNU General Public License as published by the Free Software Foundation. See the built-in help of Midnight Commander for details on the License and the lack of warranty.

AVAILABILITY

The latest version of this program can be found at <http://ftp.midnight-commander.org/>.

SEE ALSO

cooledit(1), mc(1), gpm(1), terminfo(1), scanf(3).

AUTHORS

Paul Sheer (psheer@obsidian.co.za) is the original author of Midnight Commander's internal editor.

BUGS

Bugs should be reported to <http://www.midnight-commander.org/>.

NAME

mcview – Internal file viewer of GNU Midnight Commander.

SYNOPSIS

mcview [-bcCd fhstVx?] file

DESCRIPTION

mcview is a link to **mc**, the main GNU Midnight Commander executable. Executing GNU Midnight Commander under this name requests starting the internal viewer and opening the *file* specified on the command line.

OPTIONS

- b Force black and white display.
- c Force color mode on terminals where **mcview** defaults to black and white.
- C <keyword>=<fgcolor>,<bgcolor>,<attributes>:<keyword>= ...
Specify a different color set. See the **Colors** section in mc(1) for more information.
- d Disable mouse support.
- f Display the compiled-in search paths for Midnight Commander files.
- t Used only if the code was compiled with S-Lang and terminfo: it makes Midnight Commander use the value of the **TERMCAP** variable for the terminal information instead of the information on the system wide terminal database
- V Displays the version of the program.
- x Forces xterm mode. Used when running on xterm-capable terminals (two screen modes, and able to send mouse escape sequences).

COLORS

The default colors may be changed by appending to the **MC_COLOR_TABLE** environment variable. Foreground and background colors pairs may be specified for example with:

```
MC_COLOR_TABLE="$MC_COLOR_TABLE:\
normal=lightgray,black:\
selected=black,green"
```

FILES

/usr/local/share/mc/help/mc.hlp

The help file for the program.

/usr/local/share/mc/mc.ini

The default system-wide setup for GNU Midnight Commander, used only if the user's own *~/config/mc/ini* file is missing.

/usr/local/share/mc/mc.lib

Global settings for Midnight Commander. Settings in this file affect all users, whether they have *~/config/mc/ini* or not.

~/config/mc/ini

User's own setup. If this file is present, the setup is loaded from here instead of the system-wide startup file.

LICENSE

This program is distributed under the terms of the GNU General Public License as published by the Free Software Foundation. See the built-in help of Midnight Commander for details on the License and the lack of warranty.

AVAILABILITY

The latest version of this program can be found at <http://ftp.midnight-commander.org/>.

SEE ALSO

mc(1), mcedit(1)

BUGS

Bugs should be reported to <http://www.midnight-commander.org/>.

NAME

md2html – convert Markdown to HTML

SYNOPSIS

md2html [*OPTION*]... [*FILE*]

OPTIONS**General options:**

- o, --output=*OUTFILE***
Write output to *OUTFILE* instead of **stdout**(3)
- f, --full-html**
Generate full HTML document, including header
- s, --stat**
Measure time of input parsing
- h, --help**
Display help and exit
- v, --version**
Display version and exit

Markdown dialect options:

- commonmark**
CommonMark (the default)
- github**
Github Flavored Markdown

Note: dialect options are equivalent to some combination of flags below.

Markdown extension options:

- fcollapse-whitespace**
Collapse non-trivial whitespace
- fverbatim-entities**
Do not translate entities
- fpermissive-atx-headers**
Allow ATX headers without delimiting space
- fpermissive-url-autolinks**
Allow URL autolinks without "<" and ">" delimiters
- fpermissive-www-autolinks**
Allow WWW autolinks without any scheme (e.g. "www.example.com")
- fpermissive-email-autolinks**
Allow e-mail autolinks without "<", ">" and "mailto:"
- fpermissive-autolinks**
Enable all 3 of the above permissive autolinks options
- fhard-soft-breaks**
Force all soft breaks to act as hard breaks
- fno-indented-code**
Disable indented code blocks
- fno-html-blocks**
Disable raw HTML blocks
- fno-html-spans**
Disable raw HTML spans

--fno-html

Same as **--fno-html-blocks --fno-html-spans**

--ftables

Enable tables

--fstrikethrough

Enable strikethrough spans

--ftasklists

Enable task lists

SEE ALSO

<https://github.com/mity/md4c>

NAME

md5string – filter to calculate MD5 checksum on strings

SYNOPSIS

edrc/lib/md5string [**-h** | **-V**]

stream | **md5string** [**-n**]

md5string [**-n**] < *filename*

AVAILABILITY

WA2L/edrc

DESCRIPTION

filter to calculate MD5 checksum on strings line by line.

This command is faster then using the **md5sum**(1) command in a loop if checksums for multiple strings (not files) have to be calculated.

OPTIONS

- h** usage message.
- V** print command version.
- n** no input string output beside the checksum.
- f** *filename* filename to read strings (=lines) from.

ENVIRONMENT

-

EXIT STATUS

- 0** match
- 1** cannot read from the file specified.
- 4** usage printed.

5 version printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **md5sum(1)**

NOTES

to calculate the md5 checksum for entire files, use the **md5sum(1)** command.

BUGS

-

AUTHOR

md5string was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

mediawiki – filter to get/put Wikipedia page

SYNOPSIS

edrc/lib/mediawiki [**-h**]

mediawiki **-a** { **put** | **get** } **-u** *url* **-p** *wikipage* [**-s** *section*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

get/put MediaWiki pages from/to a MediaWiki server. The communication does not need the MediaWiki API (**api.php**), therefore also MediaWiki servers running on older versions than 1.13 are supported.

OPTIONS

-h usage message.

-a action:

put
 put data stream to a MediaWiki page.

get
 get data stream from MediaWiki page and print it to **stdout**.

-u *url*
 MediaWiki URL. Example: *http://acme.ch/wiki*.

-p *page*
 MediaWiki page below the *url*.

-s *section*
 section number within the MediaWiki *page*.

ENVIRONMENT

\$MEDIAWIKI_USER

 user to connect to MediaWiki server.

\$MEDIAWIKI_PASSWD

password of connection user.

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 1 | could not log on to <i>url</i> specified with user <i>user</i> using the <i>password</i> . |
| 2 | errors while loading data to the <i>url</i> . |
| 4 | Usage printed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **vsdfml(3)**, <http://www.mediawiki.org>

NOTES

-

BUGS

-

AUTHOR

mediawiki was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

mkuser – Initialize User's Home Directory

SYNOPSIS

mkuser **-u** *uid* **-g** *gid* **-d** *directory* [**-c** *class*] [**-m** *umask*] [**-s** *server*] [**-f**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Used to initialize a user's home directory according to predefined classes. **mkuser** will not overwrite an existing directory, unless **-f** is given. The user's home directory is created, standard files, directories and symbolic links are copied, and the permissions are set. More than one user may be created, but all must be of the same class.

OPTIONS

- u** *uid* The user id of the desired user. If this is specified as a number, the creation will succeed, even if the user has not yet been created in the passwd map.

- g** *gid* The group id of the desired user. If this is specified as a number, the creation will succeed, even if the user has not yet been created in the passwd map.

- d** *directory* The physical path of the directory to be created. This is probably not the entry placed in the home directory field of the passwd entry, because the user's home directory will usually be mounted with automounter. Before it can be auto-mounted, it must physically exist.

- s** *server* Specify the server on which the homedir shall be created. It must be possible to reference the server via **/net/server**, and the parent directory must be exported with root permissions so that the directory may be created and ownership may be set.

- c** *class* If present, this option specifies the initialization "class" (or template) for the user. If specified, then files (.login, etc.) and directories are recursively copied from **~sfertools/classes/<class>** or from **\$CLASSES_DIR/classes/<class>** if the environment variable **\$CLASSES_DIR** is set. Otherwise, the directory is created and the ownership is set, but no other initialization takes place.

- f** Force initialization. Copies the standard configuration (according to -c option, if present) and sets permissions even if directory already exists. Useful if a user has deleted his "dot-files". If **mkuser** must overwrite a file, it will use the command **sav -D mkuser file** to backup the original version of **file**. It does not overwrite files which match the pattern **.my***.

- m** *umask* Set the umask for use when creating the directory. (Note: this is not applied to the users profile - this must be set in the corresponding initialization class).

ENVIRONMENT**\$CLASSES_DIR**

If this environment variable is set **mkuser** uses the classes files located in this directory for the initialization of the \$HOME of new users. See also section **FILES**.

EXIT STATUS

0 no error.

4 usage displayed.

EXAMPLES

-

SEE ALSO

edrcintro(1), **makeuser(1)**, **sav(1)**

FILES

~sfitools/classes/*

Files for initializing \$HOME of new users, may include subdirectories, which will be created.

\$CLASSES_DIR/classes/*

Files for initializing \$HOME of new users, may include subdirectories, which will be created.

~sfitools/defaults

SFI standard dot-files include the corresponding files from this subdirectory.

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

BUGS

-

AUTHOR

mkuser was developed by Peter Stevens and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

msg – write a message to screen

SYNOPSIS

edrc/lib/msg *level* "*message text*"

msg -h

AVAILABILITY

WA2L/edrc

DESCRIPTION

This command is used in scripts to write a standardized message to screen.

The message is printed to **stderr** and has the format:

<Scriptname>-<LEVEL>: <message text>

Example:

my_script-ERROR: directory '/dat/report/myreport' does not exist.

The **<Scriptname>** is derived from the **\$EDRC_SCRIPTNAME** environment variable, the **<LEVEL>** is the upper case of the *level* string given as the first option of the command. The message text is printed as received from the other options of the command.

OPTIONS

level Message level. The string given here is printed in upper case. There is no restriction in the *level* strings, a convention used in commands within WA2L/edrc is:

INFO Information messages.

ERROR

An error occurred, that questions the correct output or function of the command. An error has to be corrected to ensure that the command can complete successfully.

WARNING

A non critical malfunction of the command occurred. Often a **WARNING** can be ignored, but in some occasions a warning can also lead to a minor reduction in output quality. A warning has to be analyzed and rated by the user.

FATAL

A fatal error is a condition that has to be analyzed and that needs user intervention and correction to ensure proper functionality.

message text

message to be printed.

ENVIRONMENT

\$EDRC_SCRIPTNAME

scriptname that is printed in the **<Scriptname>** part of the output.

EXIT STATUS

0 no error.

4 usage listed.

EXAMPLES

The following examples are script cut-outs of Bourne-, Korn- or Bash shell scripts:

1) common usage

```
EDRC_SCRIPTNAME=`basename $0`; export EDRC_SCRIPTNAME
```

```
msg INFO "command started"
```

2) msg usage within a function with a return value

```
EDRC_SCRIPTNAME=`basename $0`; export EDRC_SCRIPTNAME
```

```
# is_bigger a b -- return True if a is bigger as b, else return False
#
```

```
is_bigger(){ a=$1; b=$2
    if [ $a -gt $b ]; then
        msg INFO "'$a' is bigger as '$b'"
        echo True
    else
        msg INFO "'$a' is smaller or equal as '$b'"
        echo False
    fi
}
```

```
# is_bigger
```

```
# main - main
```

```
#
```

```
main(){
    seconds=`date +%S`
    if [ `is_bigger $seconds 30` = True ]; then
        msg INFO "we are in the 2nd half of the current minute"
    else
        msg INFO "we are in the 1st half of the current minute"
    fi
}
```



```
    } # main
    main $*
```

SEE ALSO

edrcintro(1), **log(3)**, **sh(1)**, **ksh(1)**

NOTES

-

BUGS

-

AUTHOR

msg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

msmtp – An SMTP client

SYNOPSIS

Sendmail mode (default):

msmtp [option...] [--] recipient...

msmtp [option...] -t [--] [recipient...]

Server information mode:

msmtp [option...] --serverinfo

Remote Message Queue Starting mode:

msmtp [option...] --rmqs=*host*[@*domain*]*#queue*

DESCRIPTION

In the default sendmail mode, msmtp reads a mail from standard input and sends it to an SMTP server for delivery.

In server information mode, msmtp prints information about an SMTP server.

In Remote Message Queue Starting mode, msmtp sends a Remote Message Queue Starting request for a host, domain, or queue to an SMTP server.

EXIT STATUS

The standard sendmail exit status codes are used, as defined in sysexits.h.

OPTIONS

Options override configuration file settings.

They are compatible with sendmail where appropriate.

General options

--version

Print version information. This includes information about the library used for TLS/SSL support (if any), the library used for authentication, the authentication mechanisms supported by this library, and the default locations of the system and user configuration files.

--help Print help.

-P, --pretend

Print the configuration settings that would be used, but do not take further action. An asterisk (*) will be printed instead of your password.

-d, --debug

Print lots of debugging information, including the whole conversation with the SMTP server. Be careful with this option: the (potentially dangerous) output will not be sanitized, and your password may get printed in an easily decodable format!

Changing the mode of operation

-S, --serverinfo

Print information about the SMTP server and exit. This includes information about supported features (mail size limit, authentication, TLS, DSN, ...) and about the TLS certificate (if TLS is active).

--rmqs=*(host*[@*domain*]*#queue)*

Send a Remote Message Queue Starting request for the given host, domain, or queue to the SMTP server and exit.

Configuration options

-C, --file=*filename*

Use the given file instead of ~/.msmtprc as the user configuration file.

-a, --account=*account_name*

Use the given account instead of the account named "default". The settings of this account may be changed with command line options. This option cannot be used together

with the **--host** option.

--host=hostname

Use this SMTP server with settings from the command line; do not use any configuration file data. This option cannot be used together with the **--account** option.

--port=number

Set the port number to connect to. See the **port** command below.

--timeout=(off|seconds)

Set a network timeout. See the **timeout** command below. For compatibility with older versions, **--connect-timeout** is accepted as an alias for this option.

--protocol=(smtp|lmtp)

Set the protocol to use. See the **protocol** command below.

--auth[=(on|off|method)]

Enable or disable authentication. You can optionally choose the method. See the **auth** command below.

--user=[username]

Set or unset the user name for authentication. See the **user** command below.

--tls[=(on|off)]

Enable or disable TLS/SSL encryption. See the **tls** command below.

--tls-starttls[=(on|off)]

Enable or disable STARTTLS for TLS encryption. See the **tls_starttls** command below.

--tls-trust-file=[file]

Set or unset a trust file for TLS encryption. See the **tls_trust_file** command below.

--tls-crl-file=[file]

Set or unset a certificate revocation list (CRL) file for TLS. See the **tls_crl_file** command below.

--tls-key-file=[file]

Set or unset a key file for TLS encryption. See the **tls_key_file** command below.

--tls-cert-file=[file]

Set or unset a cert file for TLS encryption. See the **tls_cert_file** command below.

--tls-certcheck[=(on|off)]

Enable or disable server certificate checks for TLS encryption. See the **tls_certcheck** command below.

--tls-force-ssl3[=(on|off)]

Force TLS/SSL version SSLv3. See the **tls_force_ssl3** command below.

--tls-min-dh-prime-bits=[bits]

Set or unset minimum bit size of the Diffie-Hellman (DH) prime. See the **tls_min_dh_prime_bits** command below.

--tls-priorities=[priorities]

Set or unset TLS priorities. See the **tls_priorities** command below.

--domain=[string]

Set the argument of the SMTP EHLO (or LMTP LHLO) command. See the **domain** command below.

Options specific to sendmail mode

--auto-from[=(on|off)]

Enable or disable automatic envelope-from addresses. The default is off. See the **auto_from** command below.

- f, --from=*address*
Set the envelope-from address. It is only used when *auto_from* is off.
If no account was chosen yet (with **--account** or **--host**), this option will choose the first account that has the given envelope-from address (set with the **from** command). If no such account is found, "default" is used.
- maildomain=*domain*
Set the domain part for generated envelope-from addresses. It is only used when *auto_from* is on. See the **maildomain** command below.
- N, --dsn-notify=(*off|cond*)
Set or unset DSN notification conditions. See the **dsn_notify** command below.
- R, --dsn-return=(*off|ret*)
Set or unset the DSN notification amount. See the **dsn_return** command below. Note that *hdrs* is accepted as an alias for *headers* to be compatible with sendmail.
- keepbcc[=(*on|off*)]
Enable or disable the preservation of the Bcc header. See the **keepbcc** command below.
- X, --logfile=*file*
Set or unset the log file. See the **logfile** command below.
- syslog[=(*on|off|facility*)]
Enable or disable syslog logging. See the **syslog** command below.
- t, --read-recipients
Read recipient addresses from the To, Cc, and Bcc headers of the mail in addition to the recipients given on the command line. If any Resent- headers are present, then the addresses from any Resent-To, Resent-Cc, and Resent-Bcc headers in the first block of Resent- headers are used instead.
- read-envelope-from
Read the envelope from address from the From header of the mail.
- This marks the end of options. All following arguments will be treated as recipient addresses, even if they start with a *'*.

The following options are accepted but ignored for sendmail compatibility:

-Btype, **-bm**, **-Fname**, **-G**, **-hN**, **-i**, **-L tag**, **-m**, **-n**, **-O option=value**, **-ox value**, **-v**

USAGE

Normally, a system wide configuration file and/or a user configuration file contain information about which SMTP server to use (and how to use it), but almost all settings can also be configured on the command line.

Information about SMTP servers is organized in *accounts*. Each account describes one SMTP server: host name, authentication settings, TLS settings, and so on. Each configuration file can define multiple accounts.

In sendmail mode, an envelope-from address is necessary to send mail. This is the mail address that will be presented to the SMTP server as the originator of the mail. Envelope-from addresses can be generated automatically (when *auto_from* is enabled) or set explicitly with the **from** command or **--from** option. When *auto_from* is enabled, an envelope-from address of the form *user@domain* will be generated. The local part will be set to **USER** or, if that fails, to **LOGNAME** or, if that fails, to the login name of the current user. The domain part can be set with the **maildomain** command. If the maildomain is empty, the envelope-from address will only consist of the user name and not have a domain part.

The user can choose which account to use in one of three ways:

--account=id

Use the given account. Command line settings override configuration file settings.

`--host=hostname`

Use only the settings from the command line; do not use any configuration file data.

`--from=address` or `--read-envelope-from`

Choose the first account from the system or user configuration file that has a matching envelope-from address as specified by a **from** command. This works only when neither **--account** nor **--host** is used.

If none of the above options is used (or if no account has a matching **from** command), then the account "default" is used.

Skip to the EXAMPLES section for a quick start.

CONFIGURATION FILES

If it exists and is readable, a system wide configuration file `SYSCONFDIR/msmtpc` will be loaded, where `SYSCONFDIR` depends on your platform. Use **--version** to find out which directory is used.

If it exists and is readable, a user configuration file will be loaded (`~/msmtpc` by default). Accounts defined in the user configuration file override accounts from the system configuration file. The user configuration file must have no more permissions than user read/write. Configuration data from either file can be changed by command line options.

A configuration file is a simple text file. Empty lines and comment lines (whose first non-blank character is '#') are ignored.

Every other line must contain a command and may contain an argument to that command.

The argument may be enclosed in double quotes ("), for example if its first or last character is a blank.

If the first character of a filename is the tilde (~), this tilde will be replaced by HOME. If a command accepts the argument *on*, it also accepts an empty argument and treats that as if it was *on*.

Commands form groups. Each group begins with the **account** command and defines the settings for one SMTP server.

Skip to the EXAMPLES section for a quick start.

Commands are as follows:

defaults

Set defaults. The following configuration commands will set default values for all following account definitions in the current configuration file.

account *name* [:*account*[...]]

Start a new account definition with the given name. The current default values are filled in.

If a colon and a list of previously defined accounts is given after the account name, the new account, with the filled in default values, will inherit all settings from the accounts in the list.

host *hostname*

The SMTP server to send the mail to. The argument may be a host name or a network address. Every account definition must contain this command.

port *number*

The port that the SMTP server listens on. The default port will be acquired from your operating system's service database: for SMTP, the service is "smtp" (default port 25), unless TLS without STARTTLS is used, in which case it is "ssmtp" (465). For LMTP, it is "lmtp".

timeout (*off*|*seconds*)

Set or unset a network timeout, in seconds. The argument *off* means that no timeout will be set, which means that the operating system default will be used.

For compatibility with older versions, **connect_timeout** is accepted as an alias for this command.

protocol (*smtp*|*lmtp*)

Set the protocol to use. Currently only SMTP and LMTP are supported. SMTP is the default. See the **port** command above for default ports.

`auto_from [(on|off)]`

Enable or disable automatic envelope-from addresses. The default is off. When enabled, an envelope-from address of the form `user@domain` will be generated. The local part will be set to **USER** or, if that fails, to **LOGNAME** or, if that fails, to the login name of the current user. The domain part can be set with the **maildomain** command. If the maildomain is empty, the envelope-from address will only consist of the user name and not have a domain part. When `auto_from` is disabled, the envelope-from address must be set explicitly.

`from envelope_from`

Set the envelope-from address. This address will only be used when `auto_from` is off.

`maildomain [domain]`

Set a domain part for the generation of an envelope-from address. This is only used when `auto_from` is on. The domain may be empty.

`auth [(on|off){method}]`

This command enables or disables SMTP authentication. You should not need to set the method yourself; with the argument *on*, `msmtp` will choose the best one available for you (see below).

You probably need to set a username (with **user**) and password (with **password**). If no password is set but one is needed during authentication, `msmtp` will try to find it in `~/netrc`. If that fails, it will try to find it in `SYSCONFDIR/netrc` (use **--version** to find out what `SYSCONFDIR` is on your platform). If that fails, it will try to get it from a system specific keychain (if available). If that fails but a controlling terminal is available, `msmtp` will prompt you for it.

Currently supported keychains are the GNOME keychain and the Mac OS X keychain. See the **EXAMPLES** section below.

Available methods are *plain*, *cram-md5*, *digest-md5*, *gssapi*, *external*, *login*, and *ntlm*. Note that one or more of these methods may be unavailable due to lack of support in the underlying authentication library. Use the **--version** option to find out which methods are supported.

The *plain* and *login* methods send your authentication data in cleartext over the net, and the *ntlm* method may be vulnerable to attacks. These methods should therefore only be used together with the **tls** command.

If you don't choose the method yourself, `msmtp` chooses the best secure method that the SMTP server supports. Secure means that your authentication data will not be sent in cleartext over the net. For TLS encrypted connections, every authentication method is secure in this sense. If TLS is not active, only *gssapi*, *digest-md5*, and *cram-md5* are secure in this sense.

The *external* is special: the actual authentication happens outside of the SMTP protocol, typically by sending a TLS client certificate (see the **tls_cert_file** command). The *external* method merely confirms that this authentication succeeded for the given user (or, if no user name is given, confirms that authentication succeeded). This authentication method is not chosen automatically; you have to request it manually.

`user [username]`

Set your user name for SMTP authentication. An empty argument unsets the user name. Authentication must be activated with the **auth** command.

`password [secret]`

Set your password for SMTP authentication. An empty argument unsets the password. Authentication must be activated with the **auth** command. If no password is set but one is needed during authentication, `msmtp` will try to find it in `~/netrc`. If that fails, it will try to find it in `SYSCONFDIR/netrc` (use **--version** to find out what `SYSCONFDIR` is on your platform). If that fails, it will try to get it from a system specific keychain (if available). If that fails but a controlling terminal is available, `msmtp` will prompt you for it.

`ntlm_domain [domain]`

Set a domain for the *ntlm* authentication method. The default is to use no domain (equivalent to an empty argument), but some servers seem to require one, even if it is an arbitrary string.

tls [(on|off)]

This command enables or disables TLS (also known as SSL) encrypted connections to the SMTP server. Not every server supports TLS.

With TLS/SSL, the connection with the SMTP server will be protected against eavesdroppers and man-in-the-middle attacks. To use TLS/SSL, it is required to either use the **tls_trust_file** command (highly recommended) or to disable **tls_certcheck**.

tls_starttls [(on|off)]

By default, TLS encryption is activated using the STARTTLS SMTP command. By disabling this, TLS encryption is immediately started instead (this is known as SMTP tunneled through TLS/SSL). The default port is set to 465 for this mode of operation.

For compatibility with older versions, **tls_nostarttls** is accepted as an alias for **tls_starttls off**.

tls_trust_file [file]

This command activates strict server certificate verification.

The filename must be the absolute path name of a file in PEM format containing one or more certificates of trusted Certification Authorities (CAs).

On Debian based systems, you can install the **ca-certificates** package and use the file **/etc/ssl/certs/ca-certificates.crt**.

tls_crl_file [file]

This command sets or unsets a certificate revocation list (CRL) file for TLS, to be used during strict server certificate verification as enabled by the **tls_trust_file** command. This allows the verification procedure to detect revoked certificates.

tls_key_file [file]

This command (together with the **tls_cert_file** command) enables msmtp to send a client certificate to the SMTP server if requested. The file must contain the private key of a certificate in PEM format. An empty argument disables this feature.

tls_cert_file [file]

This command (together with the **tls_key_file** command) enables msmtp to send a client certificate to the SMTP server if requested. The file must contain a certificate in PEM format. An empty argument disables this feature.

tls_certcheck [(on|off)]

This command enables or disables checks for the server certificate.

WARNING: When the checks are disabled, TLS/SSL sessions will be vulnerable to man-in-the-middle attacks!

For compatibility with older versions, **tls_nocertcheck** is accepted as an alias for **tls_certcheck off**.

tls_force_sslv3 [(on|off)]

Force TLS/SSL version SSLv3. This might be needed to use SSL with some old and broken servers. Do not use this unless you have to.

tls_min_dh_prime_bits [bits]

Set or unset the minimum number of Diffie-Hellman (DH) prime bits that msmtp will accept for TLS sessions. The default is set by the TLS library and can be selected by using an empty argument to this command. Only lower the default (for example to 512 bits) if there is no other way to make TLS work with the remote server.

tls_priorities [priorities]

Set the priorities for TLS sessions. The default is set by the TLS library and can be selected by using an empty argument to this command. Currently this command only works with sufficiently recent GnuTLS releases. See the GnuTLS documentation of the **gnutls_priority_init** function for a description of the *priorities* string.

dsn_notify (*off*|*condition*)

This command sets the condition(s) under which the mail system should send DSN (Delivery Status Notification) messages. The argument *off* disables explicit DSN requests, which means the mail system decides when to send DSN messages. This is the default. The *condition* must be *never*, to never request notification, or a comma separated list (no spaces!) of one or more of the following: *failure*, to request notification on transmission failure, *delay*, to be notified of message delays, *success*, to be notified of successful transmission. The SMTP server must support the DSN extension.

dsn_return (*off*|*amount*)

This command controls how much of a mail should be returned in DSN (Delivery Status Notification) messages. The argument *off* disables explicit DSN requests, which means the mail system decides how much of a mail it returns in DSN messages. This is the default. The *amount* must be *headers*, to just return the message headers, or *full*, to return the full mail. The SMTP server must support the DSN extension.

domain *argument*

Use this command to set the argument of the SMTP EHLO (or LMTP LHLO) command. The default is *localhost* (stupid, but working). Possible choices are the domain part of your mail address (provider.example for joe@provider.example) or the fully qualified domain name of your host (if available).

keepbcc [(*on*|*off*)]

This command controls whether to remove or keep the Bcc header when sending a mail. The default is to remove it.

logfile [*file*]

An empty argument disables logging (this is the default).

When logging is enabled by choosing a log file, msmtp will append one line to the log file for each mail it tries to send via the account that this log file was chosen for.

The line will include the following information: date and time, host name of the SMTP server, whether TLS was used, whether authentication was used, authentication user name (only if authentication is used), envelope-from address, recipient addresses, size of the mail as transferred to the server (only if the delivery succeeded), SMTP status code and SMTP error message (only in case of failure and only if available), error message (only in case of failure and only if available), exit code (from sysexits.h; EX_OK indicates success).

If the filename is a dash (-), msmtp prints the log line to the standard output.

syslog [(*on*|*off*|*facility*)]

Enable or disable syslog logging. The facility can be one of LOG_USER, LOG_MAIL, LOG_LOCAL0, ..., LOG_LOCAL7. The default is LOG_USER.

Each time msmtp tries to send a mail via the account that contains this syslog command, it will log one entry to the syslog service with the chosen facility.

The line will include the following information: host name of the SMTP server, whether TLS was used, whether authentication was used, envelope-from address, recipient addresses, size of the mail as transferred to the server (only if the delivery succeeded), SMTP status code and SMTP error message (only in case of failure and only if available), error message (only in case of failure and only if available), exit code (from sysexits.h; EX_OK indicates success).

EXAMPLES**Configuration file**

Set default values for all following accounts.

defaults

tls on

tls_trust_file /etc/ssl/certs/ca-certificates.crt

logfile ~/.msmtp.log

A freemail service


```

account freemail
host smtp.freemail.example
from joe_smith@freemail.example
auth on
user joe.smith
password secret

# A second mail address at the same freemail service
account freemail2 : freemail
from joey@freemail.example

# The SMTP server of the provider.
account provider
host mail.provider.example
from smithjoe@provider.example
auth on
user 123456789
password my_password

# Set a default account
account default : provider

```

Manually finding the right CA certificate for `tls_trust_file`

The following example works as of 2007-04-18.

For the Gmail SMTP server, you first issue the following command:

```
msmtp --serverinfo --host=smtp.gmail.com --tls=on --port=587 --tls-certcheck=off
```

The option `--port=587` is specific to Gmail and should not be used with other servers. The option `--tls-certcheck=off` allows msmtp to accept any certificate, so that it can print some information about it.

According to the output of this command, the common name of the server certificate issuer is "Thawte Premium Server CA". This means that you have to trust the Thawte CA to use full TLS security. You can download the Thawte CA certificate bundle from <http://thawte.com/roots>. You get a ZIP file with different certificates. The one you need for the `tls_trust_file` command is *Thawte Server Roots/ThawtePremiumServerCA_b64.txt*.

The following command should now succeed:

```
msmtp --serverinfo --host=smtp.gmail.com --tls=on --port=587 --tls-trust-file="Thawte Server Roots/ThawtePremiumServerCA_b64.txt"
```

Using msmtp with Mutt

Create a configuration file for msmtp and add the following lines to your Mutt configuration file:

```

set sendmail="/path/to/msmtp"
set use_from=yes
set realname="Your Name"
set from=you@example.com
set envelope_from=yes

```

The `envelope_from=yes` option lets Mutt use the `-f` option of msmtp. Therefore msmtp chooses the first account that matches the from address `you@example.com`.

Alternatively, you can use the `-a` option:

```
set sendmail="/path/to/msmtp -a my-account"
```

Or set everything from the command line:

```
set sendmail="/path/to/msmtp --host=mailhub -f me@example.com --tls"
```

If you have multiple mail accounts in your msmtp configuration file and let Mutt use the `-f` option to choose the right one, you can easily switch accounts in Mutt with the following Mutt configuration lines:

```
macro generic "<esc>1" "":set from=you@example.com"
macro generic "<esc>2" "":set from=you@your-employer.example"
macro generic "<esc>3" "":set from=you@some-other-provider.example"
```

Using msmtp with mail

Define a default account, and put the following in your ~/.mailrc:

```
set sendmail="/path/to/msmtp"
```

Using the Mac OS X Keychain

A Mac OS X user can store a password in a keychain item using the Keychain Access GUI application. The *account name* is simply the value of the msmtp **user** argument. However, the *keychain item name* is **smtp://<hostname>** where **<hostname>** matches the msmtp **host** argument. Using **smtp://** is needed so that the item is created of kind *internet password*. For example, selecting *File->Get Info* on a keychain item that corresponds to **host smtp.freemail.example** and **user joe.smith** will show:

Name: smtp.freemail.example

Kind: Internet password

Account: joe.smith

Where: smtp://smtp.freemail.example

FILES

SYSCONFDIR/msmtpc

System configuration file. Use **--version** to find out what SYSCONFDIR is on your platform.

~/.msmtpc

User configuration file.

~/.netrc and **SYSCONFDIR/netrc**

The netrc file contains login information. If a password is not found in the configuration file, msmtp will search it in ~/.netrc and SYSCONFDIR/netrc before prompting the user for it. The syntax of netrc files is described in **netrc(5)** or **ftp(1)**.

ENVIRONMENT

USER, LOGNAME

These variables override the user's login name when constructing an envelope-from address. LOGNAME is only used if USER is unset.

TMPDIR

Directory to create temporary files in. If this is unset, a system specific default directory is used.

A temporary file is only created when the **-t/--read-recipients** or **--read-envelope-from** option is used. The file is then used to buffer the headers of the mail (but not the body, so the file won't get very large).

EMAIL, SMTPSERVER

These environment variables are used only if neither **--host** nor **--account** is used and there is no default account defined in the configuration files. In this case, the host name is taken from SMTPSERVER, and the envelope from address is taken from EMAIL, unless overridden by **--from** or **--read-envelope-from**. Currently SMTPSERVER must contain a plain host name (no URL), and EMAIL must contain a plain address (no names or additional information).

AUTHORS

msmtp was written by Martin Lambers <marlam@marlam.de>.

Other authors are listed in the AUTHORS file in the source distribution.

SEE ALSO

mutt(1), **mail(1)**, **sendmail(8)**, **netrc(5)** or **ftp(1)**

NAME

name – print description of a file or directory examined from the file header

SYNOPSIS

edrc/bin/name [*selection*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Print a description of a selection of files or directories in the current working directory examined from the file header.

This helps to get a quick overview of the purpose of files and directories located in a certain directory.

The file header must have the (very simple to remember) format:

filename - the description text

or:

path/filename - the description text

Between the *filename*, the - and *the description text* has to be only one space each.

The *filename* has to be identical to the real **filename**. See also **scripthheadersync**(1) to see how to efficiently synchronize file headers with the real file name.

If you do hard- or symbolic links to a file, each name of the file has to have an own description line in the header.

If the original file does not have a header entry of the symlinked file, the header entry of the original file is displayed.

If the file has been saved using the **sav**(3) command or if it is a backedup recovery script file as found in the **.sav** directories of a recovery script tree handled by **edrc**(1m), the **name** command can also display the header information.

If all the methods above do not succeed or if it is a directory, the index file **..name_index** is consulted for the file header information.

Binary files are not scanned for file header patterns.

OPTIONS

selection selection as known from the **ls** command to select a subset of files or directories from the current working directory.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 2 | operating system is not supported, yet. See osid (3) if you get this error. |
| 4 | usage printed. |

FILES

..name_index****

list of file or directory header descriptions for a certain directory. If this file exists and the file header cannot be resolved from the file itself the **name** command will return the matching description from that file. This might be the cause, if the file does not allow comment entries, or the file is based on a 3rd party source and it is not wished to change the file. The format is identical to the file header.

etc/name.index

global list of file header descriptions. This file is read if neither the name can be resolved from the file itself, nor from the **..**name_index**** file.

EXAMPLES

1) Shellsript header

```
#!/bin/sh
#
# busy - keep the line open
#
# [00] 28.02.2001 CWa    Initial Version
#
```

2) Configuration file header

```
#
# /etc/hosts - internet host names
#
# [00] 28.02.2001 CWa    Initial Version
#
#
127.0.0.1      localhost
192.168.75.1   rh7mzv7t001
```

3) Configuration file header

```

;
; /etc/samba/smb.conf - samba configuration file
;
; [00] 28.02.2001 CWa    Initial Version
;
;
[global]
    workgroup = MYGROUP
    server string = %h

```

4) SQL script header

```

--
-- TopTen.sql - Reports lists DB statistics
--
-- [00] 31.1.2003 CBi    Initial
--
select
    a.tablespace_name,
    a.bytes/1024 as total_kbytes,

```

5) Manpage file header

```

.\"
.\" name.1 - manpage for name
.\"
.\" [00] 30.11.2003 CWa Initial Version
.\"

```

SEE ALSO

edrcintro(1), edrcrevision(1), revision(1), scriptheadersync(1), file(1)

NOTES

-

BUGS

-

AUTHOR

name was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

nano – Nano's ANOther editor, an enhanced free Pico clone

SYNOPSIS

nano [*OPTIONS*] [[*+LINE,COLUMN*] *FILE*]...

DESCRIPTION

This manual page briefly documents the **nano** command.

nano is a small, free and friendly editor which aims to replace Pico, the default editor included in the non-free Pine package. Rather than just copying Pico's look and feel, **nano** also implements some missing (or disabled by default) features in Pico, such as "search and replace" and "go to line and column number".

OPTIONS

+LINE,COLUMN

Places cursor at line number *LINE* and column number *COLUMN* (at least one of which must be specified) on startup, instead of the default of line 1, column 1.

-? Same as **-h** (**--help**).

-A (**--smarthome**)

Make the Home key smarter. When Home is pressed anywhere but at the very beginning of non-whitespace characters on a line, the cursor will jump to that beginning (either forwards or backwards). If the cursor is already at that position, it will jump to the true beginning of the line.

-B (**--backup**)

When saving a file, back up the previous version of it to the current filename suffixed with a `~`.

-C dir (**--backupdir=dir**)

Set the directory where **nano** puts unique backup files if file backups are enabled.

-D (**--boldtext**)

Use bold text instead of reverse video text.

-E (**--tabstospaces**)

Convert typed tabs to spaces.

-F (**--multibuffer**)

Enable multiple file buffers, if available.

-H (**--historylog**)

Log search and replace strings to `%.nano_history`, so they can be retrieved in later sessions, if nanorc support is available.

-I (**--ignorercfiles**)

Don't look at `SYSCONFDIR/nanorc` or `%.nanorc`, if nanorc support is available.

-K (**--rebindkeypad**)

Interpret the numeric keypad keys so that they all work properly. You should only need to use this option if they don't, as mouse support won't work properly with this option enabled.

-L (**--nonewlines**)

Don't add newlines to the ends of files.

-N (**--noconvert**)

Disable automatic conversion of files from DOS/Mac format.

-O (**--morespace**)

Use the blank line below the titlebar as extra editing space.

- Q *str* (--quotestr=*str*)**
Set the quoting string for justifying. The default is "`^[\t]*[#:>\\|])`" if extended regular expression support is available, or ">" otherwise. Note that `\t` stands for a Tab.
- R (--restricted)**
Restricted mode: don't read or write to any file not specified on the command line; read any `nanorc` files; allow suspending; allow a file to be appended to, prepended to, or saved under a different name if it already has one; or use backup files or spell checking. Also accessible by invoking **nano** with any name beginning with 'r' (e.g. "rnano").
- S (--smooth)**
Enable smooth scrolling. Text will scroll line-by-line, instead of the usual chunk-by-chunk behavior.
- T *cols* (--tabsize=*cols*)**
Set the size (width) of a tab to *cols* columns. The value of *cols* must be greater than 0. The default value is 8.
- U (---quickblank)**
Do quick statusbar blanking. Statusbar messages will disappear after 1 keystroke instead of 25. Note that **-c** overrides this.
- V (--version)**
Show the current version number and exit.
- W (--wordbounds)**
Detect word boundaries more accurately by treating punctuation characters as part of a word.
- Y *str* (---syntax=*str*)**
Specify a specific syntax highlighting from the *nanorc* to use, if available.
- c (--const)**
Constantly show the cursor position. Note that this overrides **-U**.
- d (--rebinddelete)**
Interpret the Delete key differently so that both Backspace and Delete work properly. You should only need to use this option if Backspace acts like Delete on your system.
- h (--help)**
Show a summary of command line options and exit.
- i (--autoindent)**
Indent new lines to the previous line's indentation. Useful when editing source code.
- k (--cut)**
Enable cut from cursor to end of line.
- l (--nofollow)**
If the file being edited is a symbolic link, replace the link with a new file instead of following it. Good for editing files in */tmp*, perhaps?
- m (--mouse)**
Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute shortcuts. The mouse will work in the X Window System, and on the console when *gpm* is running.
- o *dir* (--operatingdir=*dir*)**
Set operating directory. Makes **nano** set up something similar to a chroot.
- p (--preserve)**
Preserve the XON and XOFF sequences (`^Q` and `^S`) so they will be caught by the terminal.
- r *cols* (---fill=*cols*)**
Wrap lines at column *cols*. If this value is 0 or less, wrapping will occur at the width of the screen less *cols* columns, allowing the wrap point to vary along with the width of the screen if the screen

is resized. The default value is -8.

-s *prog* (**--speller=prog**)

Enable alternative spell checker command.

-t (**--tempfile**)

Always save changed buffer without prompting. Same as Pico's **-t** option.

-v (**--view**)

View file (read only) mode.

-w (**--nowrap**)

Disable wrapping of long lines.

-x (**--nohelp**)

Disable help screen at bottom of editor.

-z (**--suspend**)

Enable suspend ability.

-a, -b, -e, -f, -g, -j

Ignored, for compatibility with Pico.

INITIALIZATION FILE

nano will read initialization files in the following order: *SYSCONFDIR/nanorc*, then *~/nanorc*. Please see **nanorc(4)** and the example file **nanorc.sample**, both of which should be provided with **nano**.

NOTES

If no alternative spell checker command is specified on the command line or in one of the nanorc files, **nano** will check the **SPELL** environment variable for one.

In some cases **nano** will try to dump the buffer into an emergency file. This will happen mainly if **nano** receives a SIGHUP or SIGTERM or runs out of memory. It will write the buffer into a file named *nano.save* if the buffer didn't have a name already, or will add a ".save" suffix to the current filename. If an emergency file with that name already exists in the current directory, it will add ".save" plus a number (e.g. ".save.1") to the current filename in order to make it unique. In multibuffer mode, **nano** will write all the open buffers to their respective emergency files.

BUGS

Please send any comments or bug reports to **nano@nano-editor.org**.

The **nano** mailing list is available from **nano-devel@gnu.org**.

To subscribe, email to **nano-devel-request@gnu.org** with a subject of "subscribe".

HOMEPAGE

<http://www.nano-editor.org/>

SEE ALSO

nanorc(4)

/usr/share/doc/nano/ (or equivalent on your system)

AUTHOR

Chris Allegretta <chrisa@asty.org>, et al (see *AUTHORS* and *THANKS* for details). This manual page was originally written by Jordi Mallach <jordi@gnu.org>, for the Debian system (but may be used by others).

NAME

nanorc – GNU nano's rcfile

DESCRIPTION

This manual page briefly documents GNU **nano**'s rcfile.

nano is a small, free and friendly editor which aims to replace Pico, the default editor included in the non-free Pine package. Rather than just copying Pico's look and feel, **nano** also implements some missing (or disabled by default) features in Pico, such as "search and replace" and "go to line and column number".

The *nanorc* file contains the default settings for **nano**. It should not be in DOS or Mac format. During startup, **nano** will first read its system-wide settings from *SYSCONFDIR/nanorc*, and then user-specific settings from *~/.nanorc*.

OPTIONS

The configuration file accepts a series of **set** and **unset** commands, which can be used to configure nano on startup without using the command line options. Additionally, the **syntax**, **color**, and **icolor** keywords are used to define syntax highlighting rules for different text patterns. **nano** will read one command per line.

Options in nanorc files take precedence over nano's defaults, and command line options override nanorc settings. Also, options are unset by default, except for those that take arguments.

Quotes inside string parameters don't have to be escaped with backslashes. The last double quote in the string will be treated as its end. For example, for the **brackets** option, `"")>]]"` will match `"`, `'`, `)`, `>`, `]`, and `]`.

The supported commands and arguments are:

set/unset autoindent

Use auto-indentation.

set/unset backup

Create backup files in *filename~*.

set backupdir *directory*

Set the directory where **nano** puts unique backup files if file backups are enabled.

set/unset backwards

Do backwards searches by default.

set/unset boldtext

Use bold text instead of reverse video text.

set brackets *string*

Set the characters treated as closing brackets when justifying paragraphs. They cannot contain blank characters. Only closing punctuation, optionally followed by closing brackets, can end sentences. The default value is `"")>]]"`.

set/unset casesensitive

Do case sensitive searches by default.

set/unset const

Constantly display the cursor position in the status bar.

set/unset cut

Use cut to end of line by default.

set fill *n*

Wrap lines at column number *n*. If *n* is 0 or less, the maximum line length will be the screen width less *n* columns. The default value is -8.

set/unset historylog

Enable `~/.nano_history` for saving and reading search/replace strings.

set matchbrackets *string*

Set the opening and closing brackets that can be found by bracket searches. They cannot contain blank characters. The former set must come before the latter set, and both must be in the same order. The default value is "`(<[{}>])`".

set/unset morespace

Use the blank line below the titlebar as extra editing space.

set/unset mouse

Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute shortcuts. The mouse will work in the X Window System, and on the console when gpm is running.

set/unset multibuffer

Allow inserting files into their own buffers.

set/unset noconvert

Don't convert files from DOS/Mac format.

set/unset nofollow

Don't follow symlinks when writing files.

set/unset nohelp

Don't display the help lists at the bottom of the screen.

set/unset nonewlines

Don't add newlines to the ends of files.

set/unset nowrap

Don't wrap text at all.

set operatingdir *directory*

nano will only read and write files inside *directory* and its subdirectories. Also, the current directory is changed to here, so files are inserted from this directory. By default, the operating directory feature is turned off.

set/unset preserve

Preserve the XON and XOFF keys (^Q and ^S).

set punct *string*

Set the characters treated as closing punctuation when justifying paragraphs. They cannot contain blank characters. Only closing punctuation, optionally followed by closing brackets, can end sentences. The default value is "`!.?>`".

set/unset quickblank

Do quick statusbar blanking. Statusbar messages will disappear after 1 keystroke instead of 25.

set quotestr *string*

The email-quote string, used to justify email-quoted paragraphs. This is an extended regular expression if your system supports them, otherwise a literal string. The default value is "`^[\t]*[#:]>[\t]*`" if you have extended regular expression support, or "`>`" otherwise. Note that `\t` stands for a literal Tab character.

set/unset rebinddelete

Interpret the Delete key differently so that both Backspace and Delete work properly. You should only need to use this option if Backspace acts like Delete on your system.

set/unset rebindkeypad

Interpret the numeric keypad keys so that they all work properly. You should only need to use this option if they don't, as mouse support won't work properly with this option enabled.

set/unset regexp

Do extended regular expression searches by default.

set/unset smarthome

Make the Home key smarter. When Home is pressed anywhere but at the very beginning of non-white-space characters on a line, the cursor will jump to that beginning (either forwards or backwards). If the cursor is already at that position, it will jump to the true beginning of the line.

set/unset smooth

Use smooth scrolling by default.

set speller *spellprog*

Use spelling checker *spellprog* instead of the built-in one, which calls *spell*.

set/unset suspend

Allow **nano** to be suspended.

set tabsize *n*

Use a tab size of *n* columns. The value of *n* must be greater than 0. The default value is 8.

set/unset tabstospaces

Convert typed tabs to spaces.

set/unset tempfile

Save automatically on exit, don't prompt.

set/unset view

Disallow file modification.

set whitespace *string*

Set the two characters used to display the first characters of tabs and spaces. They must be single-column characters.

set/unset wordbounds

Detect word boundaries more accurately by treating punctuation characters as parts of words.

syntax *str* ["*fileregex*" ...]

Defines a syntax named *str* which can be activated via the -Y/--syntax command line option, or will be automatically activated if the current filename matches the extended regular expression *fileregex*. All following **color** and **icolor** statements will apply to *syntax* until a new syntax is defined.

The *none* syntax is reserved; specifying it on the command line is the same as not having a syntax at all. The *default* syntax is special: it takes no *fileregex*, and applies to files that don't match any other syntax's *fileregex*.

color *fgcolor,bgcolor regex* ...

For the currently defined syntax, display all expressions matching the extended regular expression *regex* with foreground color *fgcolor* and background color *bgcolor*, at least one of which must be specified. Legal colors for foreground and background color are: white, black, red, blue, green, yellow, magenta, and cyan. You may use the prefix "bright" to force a stronger color highlight for the foreground. If your terminal supports transparency, not specifying a *bgcolor* tells **nano** to attempt to use a transparent background.

icolor *fgcolor,bgcolor regex* ...

Same as above, except that the expression matching is case insensitive.

color *fgcolor,bgcolor start="sr" end="er"*

Display expressions which start with the extended regular expression *sr* and end with the extended regular expression *er* with foreground color *fgcolor* and background color *bgcolor*, at least one of which must be specified. This allows syntax highlighting to span multiple lines. Note that all subsequent instances of *sr* after an initial *sr* is found will be highlighted until the first instance of *er*.

icolor *fgcolor,bgcolor* **start="sr" end="er"**

Same as above, except that the expression matching is case insensitive.

include *syntaxfile*

Read in self-contained color syntaxes from *syntaxfile*. Note that *syntaxfile* can only contain **syntax**, **color**, and **icolor** commands.

FILES

SYSCONFDIR/nanorc

System-wide configuration file

~/.nanorc

Per-user configuration file

SEE ALSO

nano(1)

/usr/share/doc/nano/examples/nanorc.sample (or equivalent on your system)

AUTHOR

Chris Allegretta <chrisa@asty.org>, et al (see *AUTHORS* and *THANKS* for details). This manual page was originally written by Jordi Mallach <jordi@gnu.org>, for the Debian system (but may be used by others).

NAME

nc – GNU Netcat Manual

SYNOPSIS

nc [options] hostname port [port] ...

nc -l -p port [options] [hostname] [port] ...

nc -L hostname:port -p port [options] ...

DESCRIPTION

nc is a simple Unix utility which reads and writes data across network connections, using TCP or UDP protocol. It is designed to be a reliable “back-end” tool that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and exploration tool, since it can create almost any kind of connection you would need and has several interesting built-in capabilities. nc, or “nc” as the original program was named, should have been supplied long ago as another one of those cryptic but standard Unix tools.

nc has three main modes of functionality. These are the connect mode, the listen mode, and the tunnel mode.

The most common mode is the connect mode, which for example allows the output of a locally called command to be redirected for example to a remote nc listening or to any other kind of daemon waiting for a connection.

On the other hand, the listen mode can be used to obtain some kind of stream of data from a remote site.

The most new feature is the tunnel mode, which is a powerful and reliable mode that allows tunneling a remote site towards any other remote site, allowing to specify for example from which interface create the connection and from which port.

OPTIONS**Basic Startup Options**

-V

--version

Display the version of nc and exit.

-h

--help

Print a help message describing most common nc’s command-line switches and a short description.

-v

--verbose

Prints status messages, usually needed for using nc as user front-end. All messages are printed to stderr in order not to affect the data stream.

Use this option double to get more messages.

Protocol and Interface Options

-t

--tcp

Selects the TCP protocol, this is the default. It may be useful (see Tunnel Mode) to specify this option after for example the UDP option in order to allow a cross-protocol bridge between TCP and UDP.

-u

--udp

Selects the UDP protocol. See the **--tcp** option.

-p NUM

--local-port=NUM

Selects the local port. In listen and tunnel mode, it specifies which port to use for listening, while in connect mode it specifies the source port (the port from which originating the connection).

If this option is not specified, the OS will assign a random available port.

-s ADDRESS

--source=ADDRESS

Specifies the source address used for creating sockets. In listen mode and tunnel mode this switch specifies the bound address, and it is generally a good idea not to specify this, which causes nc to bind to a generic interface. In the connect mode, this switch is used to specify the source address for connecting to the outside world. Again, if it's not specified a proper address for the destination route will be used.

-P NUM

--tunnel-port=NUM

Same as **--port**, but affects only the connect phase (thus this option has no effect in listen mode). This switch is useful in tunnel mode for specifying the source port for the connecting socket.

-S ADDRESS

--tunnel-source=ADDRESS

Same as **--source**, but affects only the connect phase (thus this has no effects in listen mode). This switch is useful in tunnel mode for specifying the source address for the connecting socket.

Advanced Options

-i SECS

--interval SECS

sets the buffering output delay time. This affects all the current modes and makes the connection sock to buffer outgoing data. This means that in tunnel mode everything received from the listening socket is buffered for the connect socket.

-n

--dont-resolve

Don't do DNS lookups on any of the specified addresses or hostnames, or names of port numbers from `/etc/services`.

-r

--randomize

Randomizes the target remote ports ranges. If more than one range is specified it will randomize the ports in the whole global range.

-w

--wait=SECS

Specifies the starting inactivity delay after which nc will exit with an error status. In connect mode and in tunnel mode this specifies the timeout for the connecting socket, while in listen mode it specifies the time to wait for a VALID incoming connection (see listen mode).

-T

--telnet

Answers the telnet codes as described in RFC0854. This makes possible to use nc to script telnet sessions. The incoming telnet codes are parsed inside the receiving queue and are stripped off before forwarding the data as they were never received, so the application doesn't have to parse the codes itself (this behaviour can be disabled at compile time with **--enable-oldtelnet** or with **--enable-compat**).

-z

--zero

Sets the zero I/O flag for the selected mode. In connect mode it means that as soon as the port is open it is immediately shutdown and closed. This may be useful for probing or scanning (even if there are faster portscanners out there, but this may be useful for scripting purposes). In listen mode, it makes nc refusing all the incoming connections thus running in timeout (if set), or waiting forever. In both cases, no data is transferred.

This option is incompatible with the tunnel mode.

SEE ALSO

GNU Info entry for *nc*.

AUTHOR

Originally written by Giovanni Giacobbi <giovanni@giacobbi.net>.

COPYRIGHT

Copyright (c) 2002 – 2004 Giovanni Giacobbi

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being “GNU General Public License” and “GNU Free Documentation License”, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

NAME

nginx - HTTP and reverse proxy server, mail proxy server

SYNOPSIS

lib/nginx [**-?hqtVv**] [**-c** *file*] [**-g** *directives*] [**-p** *prefix*] [**-s** *signal*]

DESCRIPTION

nginx (pronounced engine x) is an HTTP and reverse proxy server, as well as a mail proxy server. It is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

OPTIONS

-h, -? Print help.

-c *file* Use an alternative configuration *file* .

-g *directives*
Set global configuration directives. See **EXAMPLES** for details.

-p *prefix* Set the prefix path. The default value is `%%PREFIX%%` .

-q Suppress non-error messages during configuration testing.

-s *signal* Send a signal to the master process. The argument *signal* can be one of: **stop** , **quit** , **reopen** , **reload** .

The following table shows the corresponding system signals:

```
stop SIGTERM
quit SIGQUIT
reopen SIGUSR1
reload SIGHUP
```

-t Do not run, just test the configuration file. **nginx** checks the configuration file syntax and then tries to open files referenced in the configuration file.

-V Print the **nginx** version, compiler version, and *configure* script parameters.

-v Print the **nginx** version.

SIGNALS

The master process of **nginx** can handle the following signals:

SIGINT, SIGTERM

Shut down quickly.

SIGHUP Reload configuration, start the new worker process with a new configuration, and gracefully shut down old worker processes.

SIGQUIT Shut down gracefully.

SIGUSR1 Reopen log files.

SIGUSR2 Upgrade the **nginx** executable on the fly.

SIGWINCH

Shut down worker processes gracefully.

While there is no need to explicitly control worker processes normally, they support some signals too:

SIGTERM

Shut down quickly.

SIGQUIT Shut down gracefully.

SIGUSR1 Reopen log files.

DEBUGGING LOG

To enable a debugging log, reconfigure **nginx** to build with debugging:

```
./configure --with-debug ...
```

and then set the **debug** level of the **error_log** :

```
error_log /path/to/log debug;
```

It is also possible to enable the debugging for a particular IP address:

```
events {
    debug_connection 127.0.0.1;
}
```

ENVIRONMENT

The **\$NGINX** environment variable is used internally by **nginx** and should not be set directly by the user.

FILES

%%PID_PATH%

Contains the process ID of **nginx** . The contents of this file are not sensitive, so it can be world-readable.

%%CONF_PATH%

The main configuration file.

%%ERROR_LOG_PATH%

Error log file.

EXIT STATUS

Exit status is **0** on success, or **1** if the command fails.

EXAMPLES

Test configuration file **~/mynginx.cfg** with global directives for PID and quantity of worker processes:

```
nginx -t -c ~/mynginx.conf \  
-g "pid /var/run/mynginx.pid; worker_processes 2;"
```

SEE ALSO

Documentation at <http://nginx.org/en/docs/> .

For questions and technical support, please refer to <http://nginx.org/en/support.html> .

HISTORY

Development of **nginx** started in 2002, with the first public release on October 4, 2004.

AUTHORS

Igor Sysoev <igor@sysoev.ru> .

This manual page was originally written by Sergey A. Osokin <osa@FreeBSD.org.ru> as a result of compiling many **nginx** documents from all over the world.

NAME

nmap – Network exploration tool and security / port scanner

SYNOPSIS

nmap [*Scan Type...*] [*Options*] {*target specification*}

DESCRIPTION

Nmap (Network Mapper) is an open source tool for network exploration and security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. While Nmap is commonly used for security audits, many systems and network administrators find it useful for routine tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime.

The output from Nmap is a list of scanned targets, with supplemental information on each depending on the options used. Key among that information is the “interesting ports table”.. That table lists the port number and protocol, service name, and state. The state is either open, filtered, closed, or unfiltered. Open. means that an application on the target machine is listening for connections/packets on that port. Filtered. means that a firewall, filter, or other network obstacle is blocking the port so that Nmap cannot tell whether it is open or closed. Closed. ports have no application listening on them, though they could open up at any time. Ports are classified as unfiltered. when they are responsive to Nmap's probes, but Nmap cannot determine whether they are open or closed. Nmap reports the state combinations open|filtered. and closed|filtered. when it cannot determine which of the two states describe a port. The port table may also include software version details when version detection has been requested. When an IP protocol scan is requested (**-sO**), Nmap provides information on supported IP protocols rather than listening ports.

In addition to the interesting ports table, Nmap can provide further information on targets, including reverse DNS names, operating system guesses, device types, and MAC addresses.

A typical Nmap scan is shown in Example 1. The only Nmap arguments used in this example are **-A**, to enable OS and version detection, script scanning, and traceroute; **-T4** for faster execution; and then the two target hostnames.

Example 1. A representative Nmap scan

```
# nmap -A -T4 scanme.nmap.org
```

```
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
| ssh-hostkey: 1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (DSA)
|_ 2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)
80/tcp    open  http     Apache httpd 2.2.14 ((Ubuntu))
|_ http-title: Go ahead and ScanMe!
646/tcp   filtered ldap
1720/tcp   filtered H.323/Q.931
9929/tcp   open  nping-echo Nping echo
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
OS details: Linux 2.6.39
Network Distance: 11 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:kernel
```

```

TRACEROUTE (using port 53/tcp)
HOP RTT    ADDRESS
[Cut first 10 hops for brevity]
11  17.65 ms li86-221.members.linode.com (74.207.244.221)

```

Nmap done: 1 IP address (1 host up) scanned in 14.40 seconds

The newest version of Nmap can be obtained from <http://nmap.org>. The newest version of this man page is available at <http://nmap.org/book/man.html>. It is also included as a chapter of Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning (see <http://nmap.org/book/>).

OPTIONS SUMMARY

This options summary is printed when Nmap is run with no arguments, and the latest version is always available at <https://svn.nmap.org/nmap/docs/nmap.usage.txt>. It helps people remember the most common options, but is no substitute for the in-depth documentation in the rest of this manual. Some obscure options aren't even included here.

Nmap 6.25 (<http://nmap.org>)

Usage: nmap [Scan Type(s)] [Options] {target specification}

TARGET SPECIFICATION:

Can pass hostnames, IP addresses, networks, etc.

Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

-iL <inputfilename>: Input from list of hosts/networks

-iR <num hosts>: Choose random targets

--exclude <host1[,host2][,host3],...>: Exclude hosts/networks

--excludefile <exclude_file>: Exclude list from file

HOST DISCOVERY:

-sL: List Scan - simply list targets to scan

-sn: Ping Scan - disable port scan

-Pn: Treat all hosts as online -- skip host discovery

-PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports

-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes

-PO[protocol list]: IP Protocol Ping

-n/-R: Never do DNS resolution/Always resolve [default: sometimes]

--dns-servers <serv1[,serv2],...>: Specify custom DNS servers

--system-dns: Use OS's DNS resolver

--traceroute: Trace hop path to each host

SCAN TECHNIQUES:

-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans

-sU: UDP Scan

-sN/sF/sX: TCP Null, FIN, and Xmas scans

--scanflags <flags>: Customize TCP scan flags

-sI <zombie host[:probeport]>: Idle scan

-sY/sZ: SCTP INIT/COOKIE-ECHO scans

-sO: IP protocol scan

-b <FTP relay host>: FTP bounce scan

PORT SPECIFICATION AND SCAN ORDER:

-p <port ranges>: Only scan specified ports

Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9

-F: Fast mode - Scan fewer ports than the default scan

-r: Scan ports consecutively - don't randomize

--top-ports <number>: Scan <number> most common ports

--port-ratio <ratio>: Scan ports more common than <ratio>

SERVICE/VERSION DETECTION:

-sV: Probe open ports to determine service/version info

--version-intensity <level>: Set from 0 (light) to 9 (try all probes)
 --version-light: Limit to most likely probes (intensity 2)
 --version-all: Try every single probe (intensity 9)
 --version-trace: Show detailed version scan activity (for debugging)

SCRIPT SCAN:

-sC: equivalent to --script=default
 --script=<Lua scripts>: <Lua scripts> is a comma separated list of directories, script-files or script-categories
 --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
 --script-args-file=filename: provide NSE script args in a file
 --script-trace: Show all data sent and received
 --script-updatedb: Update the script database.
 --script-help=<Lua scripts>: Show help about scripts.
 <Lua scripts> is a comma separated list of script-files or script-categories.

OS DETECTION:

-O: Enable OS detection
 --osscan-limit: Limit OS detection to promising targets
 --osscan-guess: Guess OS more aggressively

TIMING AND PERFORMANCE:

Options which take <time> are in seconds, or append 'ms' (milliseconds), 's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
 -T<0-5>: Set timing template (higher is faster)
 --min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
 --min-parallelism/max-parallelism <numprobes>: Probe parallelization
 --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies probe round trip time.
 --max-retries <tries>: Caps number of port scan probe retransmissions.
 --host-timeout <time>: Give up on target after this long
 --scan-delay/--max-scan-delay <time>: Adjust delay between probes
 --min-rate <number>: Send packets no slower than <number> per second
 --max-rate <number>: Send packets no faster than <number> per second

FIREWALL/IDS EVASION AND SPOOFING:

-f; --mtu <val>: fragment packets (optionally w/given MTU)
 -D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
 -S <IP_Address>: Spoof source address
 -e <iface>: Use specified interface
 -g/--source-port <portnum>: Use given port number
 --data-length <num>: Append random data to sent packets
 --ip-options <options>: Send packets with specified ip options
 --ttl <val>: Set IP time-to-live field
 --spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
 --badsum: Send packets with a bogus TCP/UDP/SCTP checksum

OUTPUT:

-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rlpt klddi3, and Grepable format, respectively, to the given filename.
 -oA <basename>: Output in the three major formats at once
 -v: Increase verbosity level (use -vv or more for greater effect)
 -d: Increase debugging level (use -dd or more for greater effect)
 --reason: Display the reason a port is in a particular state
 --open: Only show open (or possibly open) ports
 --packet-trace: Show all packets sent and received
 --iflist: Print host interfaces and routes (for debugging)
 --log-errors: Log errors/warnings to the normal-format output file

--append-output: Append to rather than clobber specified output files
 --resume <filename>: Resume an aborted scan
 --stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
 --webxml: Reference stylesheet from Nmap.Org for more portable XML
 --no-stylesheet: Prevent associating of XSL stylesheet w/XML output

MISC:

-6: Enable IPv6 scanning
 -A: Enable OS detection, version detection, script scanning, and traceroute
 --datadir <dirname>: Specify custom Nmap data file location
 --send-eth/--send-ip: Send using raw ethernet frames or IP packets
 --privileged: Assume that the user is fully privileged
 --unprivileged: Assume the user lacks raw socket privileges
 -V: Print version number
 -h: Print this help summary page.

EXAMPLES:

```
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -iR 10000 -Pn -p 80
```

SEE THE MAN PAGE (<http://nmap.org/book/man.html>) FOR MORE OPTIONS AND EXAMPLES

TARGET SPECIFICATION

Everything on the Nmap command-line that isn't an option (or option argument) is treated as a target host specification. The simplest case is to specify a target IP address or hostname for scanning.

Sometimes you wish to scan a whole network of adjacent hosts. For this, Nmap supports CIDR-style addressing. You can append */numbits* to an IPv4 address or hostname and Nmap will scan every IP address for which the first *numbits* are the same as for the reference IP or hostname given. For example, 192.168.10.0/24 would scan the 256 hosts between 192.168.10.0 (binary: 11000000 10101000 00001010 00000000) and 192.168.10.255 (binary: 11000000 10101000 00001010 11111111), inclusive. 192.168.10.40/24 would scan exactly the same targets. Given that the host scanme.nmap.org. is at the IP address 64.13.134.52, the specification scanme.nmap.org/16 would scan the 65,536 IP addresses between 64.13.0.0 and 64.13.255.255. The smallest allowed value is /0, which targets the whole Internet. The largest value is /32, which scans just the named host or IP address because all address bits are fixed.

CIDR notation is short but not always flexible enough. For example, you might want to scan 192.168.0.0/16 but skip any IPs ending with .0 or .255 because they may be used as subnet network and broadcast addresses. Nmap supports this through octet range addressing. Rather than specify a normal IP address, you can specify a comma-separated list of numbers or ranges for each octet. For example, 192.168.0-255.1-254 will skip all addresses in the range that end in .0 or .255, and 192.168.3-5.7.1 will scan the four addresses 192.168.3.1, 192.168.4.1, 192.168.5.1, and 192.168.7.1. Either side of a range may be omitted; the default values are 0 on the left and 255 on the right. Using - by itself is the same as 0-255, but remember to use 0- in the first octet so the target specification doesn't look like a command-line option. Ranges need not be limited to the final octets: the specifier 0-255.0-255.13.37 will perform an Internet-wide scan for all IP addresses ending in 13.37. This sort of broad sampling can be useful for Internet surveys and research.

IPv6 addresses can only be specified by their fully qualified IPv6 address or hostname. CIDR and octet ranges aren't yet supported for IPv6.

IPv6 addresses with non-global scope need to have a zone ID suffix. On Unix systems, this is a percent sign followed by an interface name; a complete address might be fe80::a8bb:ccff:fedd:eeff%eth0. On Windows, use an interface index number in place of an interface name: fe80::a8bb:ccff:fedd:eeff%1. You can see a list of interface indexes by running the command **netsh.exe interface ipv6 show interface**.

Nmap accepts multiple host specifications on the command line, and they don't need to be the same type. The command **nmap scanme.nmap.org 192.168.0.0/8 10.0.0.1,3-7.-** does what you would expect.

While targets are usually specified on the command lines, the following options are also available to control target selection:

-iL *inputfilename* (Input from list) .

Reads target specifications from *inputfilename*. Passing a huge list of hosts is often awkward on the command line, yet it is a common desire. For example, your DHCP server might export a list of 10,000 current leases that you wish to scan. Or maybe you want to scan all IP addresses *except* for those to locate hosts using unauthorized static IP addresses. Simply generate the list of hosts to scan and pass that filename to Nmap as an argument to the **-iL** option. Entries can be in any of the formats accepted by Nmap on the command line (IP address, hostname, CIDR, IPv6, or octet ranges). Each entry must be separated by one or more spaces, tabs, or newlines. You can specify a hyphen (-) as the filename if you want Nmap to read hosts from standard input rather than an actual file.

The input file may contain comments that start with # and extend to the end of the line.

-iR *num hosts* (Choose random targets) .

For Internet-wide surveys and other research, you may want to choose targets at random. The *num hosts* argument tells Nmap how many IPs to generate. Undesirable IPs such as those in certain private, multicast, or unallocated address ranges are automatically skipped. The argument 0 can be specified for a never-ending scan. Keep in mind that some network administrators bristle at unauthorized scans of their networks and may complain. Use this option at your own risk! If you find yourself really bored one rainy afternoon, try the command **nmap -Pn -sS -p 80 -iR 0 --open** to locate random web servers for browsing.

--exclude *host1[,host2[,...]]* (Exclude hosts/networks) .

Specifies a comma-separated list of targets to be excluded from the scan even if they are part of the overall network range you specify. The list you pass in uses normal Nmap syntax, so it can include hostnames, CIDR netblocks, octet ranges, etc. This can be useful when the network you wish to scan includes untouchable mission-critical servers, systems that are known to react adversely to port scans, or subnets administered by other people.

--excludefile *exclude_file* (Exclude list from file) .

This offers the same functionality as the **--exclude** option, except that the excluded targets are provided in a newline-, space-, or tab-delimited *exclude_file* rather than on the command line.

The exclude file may contain comments that start with # and extend to the end of the line.

HOST DISCOVERY

One of the very first steps in any network reconnaissance mission is to reduce a (sometimes huge) set of IP ranges into a list of active or interesting hosts. Scanning every port of every single IP address is slow and usually unnecessary. Of course what makes a host interesting depends greatly on the scan purposes. Network administrators may only be interested in hosts running a certain service, while security auditors may care about every single device with an IP address. An administrator may be comfortable using just an ICMP ping to locate hosts on his internal network, while an external penetration tester may use a diverse set of dozens of probes in an attempt to evade firewall restrictions.

Because host discovery needs are so diverse, Nmap offers a wide variety of options for customizing the techniques used. Host discovery is sometimes called ping scan, but it goes well beyond the simple ICMP echo request packets associated with the ubiquitous ping tool. Users can skip the ping step entirely with a list scan (**-sL**) or by disabling ping (**-Pn**), or engage the network with arbitrary combinations of multi-port TCP SYN/ACK, UDP, SCTP INIT and ICMP probes. The goal of these probes is to solicit responses which demonstrate that an IP address is actually active (is being used by a host or network device). On many networks, only a small percentage of IP addresses are active at any given time. This is particularly common with private address space such as 10.0.0.0/8. That network has 16 million IPs, but I have seen it used by companies with less than a thousand machines. Host discovery can find those machines in a sparsely allocated sea of IP addresses.

If no host discovery options are given, Nmap sends an ICMP echo request, a TCP SYN packet to port 443, a TCP ACK packet to port 80, and an ICMP timestamp request. (For IPv6, the ICMP timestamp request is omitted because it is not part of ICMPv6.) These defaults are equivalent to the **-PE -PS443 -PA80 -PP** options. The exceptions to this are the ARP (for IPv4) and Neighbor Discovery. (for IPv6) scans which are

used for any targets on a local ethernet network. For unprivileged Unix shell users, the default probes are a SYN packet to ports 80 and 443 using the **connect** system call.. This host discovery is often sufficient when scanning local networks, but a more comprehensive set of discovery probes is recommended for security auditing.

The **-P*** options (which select ping types) can be combined. You can increase your odds of penetrating strict firewalls by sending many probe types using different TCP ports/flags and ICMP codes. Also note that ARP/Neighbor Discovery (**-PR**) is done by default against targets on a local ethernet network even if you specify other **-P*** options, because it is almost always faster and more effective.

By default, Nmap does host discovery and then performs a port scan against each host it determines is online. This is true even if you specify non-default host discovery types such as UDP probes (**-PU**). Read about the **-sn** option to learn how to perform only host discovery, or use **-Pn** to skip host discovery and port scan all target hosts. The following options control host discovery:

-sL (List Scan) .

The list scan is a degenerate form of host discovery that simply lists each host of the network(s) specified, without sending any packets to the target hosts. By default, Nmap still does reverse-DNS resolution on the hosts to learn their names. It is often surprising how much useful information simple hostnames give out. For example, fw.chi is the name of one company's Chicago firewall. Nmap also reports the total number of IP addresses at the end. The list scan is a good sanity check to ensure that you have proper IP addresses for your targets. If the hosts sport domain names you do not recognize, it is worth investigating further to prevent scanning the wrong company's network.

Since the idea is to simply print a list of target hosts, options for higher level functionality such as port scanning, OS detection, or ping scanning cannot be combined with this. If you wish to disable ping scanning while still performing such higher level functionality, read up on the **-Pn** (skip ping) option.

-sn (No port scan) .

This option tells Nmap not to do a port scan after host discovery, and only print out the available hosts that responded to the scan. This is often known as a "ping scan", but you can also request that traceroute and NSE host scripts be run. This is by default one step more intrusive than the list scan, and can often be used for the same purposes. It allows light reconnaissance of a target network without attracting much attention. Knowing how many hosts are up is more valuable to attackers than the list provided by list scan of every single IP and host name.

Systems administrators often find this option valuable as well. It can easily be used to count available machines on a network or monitor server availability. This is often called a ping sweep, and is more reliable than pinging the broadcast address because many hosts do not reply to broadcast queries.

The default host discovery done with **-sn** consists of an ICMP echo request, TCP SYN to port 443, TCP ACK to port 80, and an ICMP timestamp request by default. When executed by an unprivileged user, only SYN packets are sent (using a **connect** call) to ports 80 and 443 on the target. When a privileged user tries to scan targets on a local ethernet network, ARP requests are used unless **--send-ip** was specified. The **-sn** option can be combined with any of the discovery probe types (the **-P*** options, excluding **-Pn**) for greater flexibility. If any of those probe type and port number options are used, the default probes are overridden. When strict firewalls are in place between the source host running Nmap and the target network, using those advanced techniques is recommended. Otherwise hosts could be missed when the firewall drops probes or their responses.

In previous releases of Nmap, **-sn** was known as **-sP**..

-Pn (No ping) .

This option skips the Nmap discovery stage altogether. Normally, Nmap uses this stage to determine active machines for heavier scanning. By default, Nmap only performs heavy probing such as port scans, version detection, or OS detection against hosts that are found to be up. Disabling host discovery with **-Pn** causes Nmap to attempt the requested scanning functions against *every* target IP

address specified. So if a class B target address space (/16) is specified on the command line, all 65,536 IP addresses are scanned. Proper host discovery is skipped as with the list scan, but instead of stopping and printing the target list, Nmap continues to perform requested functions as if each target IP is active. To skip ping scan *and* port scan, while still allowing NSE to run, use the two options **-Pn** **-sn** together.

For machines on a local ethernet network, ARP scanning will still be performed (unless **--disable-arp-ping** or **--send-ip** is specified) because Nmap needs MAC addresses to further scan target hosts. In previous versions of Nmap, **-Pn** was **-P0**. and **-PN**..

-PS *port list* (TCP SYN Ping) .

This option sends an empty TCP packet with the SYN flag set. The default destination port is 80 (configurable at compile time by changing `DEFAULT_TCP_PROBE_PORT_SPEC`. in `nmap.h`).. Alternate ports can be specified as a parameter. The syntax is the same as for the **-p** except that port type specifiers like T: are not allowed. Examples are **-PS22** and **-PS22-25,80,113,1050,35000**. Note that there can be no space between **-PS** and the port list. If multiple probes are specified they will be sent in parallel.

The SYN flag suggests to the remote system that you are attempting to establish a connection. Normally the destination port will be closed, and a RST (reset) packet sent back. If the port happens to be open, the target will take the second step of a TCP three-way-handshake. by responding with a SYN/ACK TCP packet. The machine running Nmap then tears down the nascent connection by responding with a RST rather than sending an ACK packet which would complete the three-way-handshake and establish a full connection. The RST packet is sent by the kernel of the machine running Nmap in response to the unexpected SYN/ACK, not by Nmap itself.

Nmap does not care whether the port is open or closed. Either the RST or SYN/ACK response discussed previously tell Nmap that the host is available and responsive.

On Unix boxes, only the privileged user root. is generally able to send and receive raw TCP packets.. For unprivileged users, a workaround is automatically employed. whereby the **connect** system call is initiated against each target port. This has the effect of sending a SYN packet to the target host, in an attempt to establish a connection. If **connect** returns with a quick success or an ECONNREFUSED failure, the underlying TCP stack must have received a SYN/ACK or RST and the host is marked available. If the connection attempt is left hanging until a timeout is reached, the host is marked as down.

-PA *port list* (TCP ACK Ping) .

The TCP ACK ping is quite similar to the just-discussed SYN ping. The difference, as you could likely guess, is that the TCP ACK flag is set instead of the SYN flag. Such an ACK packet purports to be acknowledging data over an established TCP connection, but no such connection exists. So remote hosts should always respond with a RST packet, disclosing their existence in the process.

The **-PA** option uses the same default port as the SYN probe (80) and can also take a list of destination ports in the same format. If an unprivileged user tries this, the **connect** workaround discussed previously is used. This workaround is imperfect because **connect** is actually sending a SYN packet rather than an ACK.

The reason for offering both SYN and ACK ping probes is to maximize the chances of bypassing firewalls. Many administrators configure routers and other simple firewalls to block incoming SYN packets except for those destined for public services like the company web site or mail server. This prevents other incoming connections to the organization, while allowing users to make unobstructed outgoing connections to the Internet. This non-stateful approach takes up few resources on the firewall/router and is widely supported by hardware and software filters. The Linux Netfilter/iptables. firewall software offers the **--syn** convenience option to implement this stateless approach. When stateless firewall rules such as this are in place, SYN ping probes (**-PS**) are likely to be blocked when

sent to closed target ports. In such cases, the ACK probe shines as it cuts right through these rules.

Another common type of firewall uses stateful rules that drop unexpected packets. This feature was initially found mostly on high-end firewalls, though it has become much more common over the years. The Linux Netfilter/iptables system supports this through the **--state** option, which categorizes packets based on connection state. A SYN probe is more likely to work against such a system, as unexpected ACK packets are generally recognized as bogus and dropped. A solution to this quandary is to send both SYN and ACK probes by specifying **-PS** and **-PA**.

-PU *port list* (UDP Ping) .

Another host discovery option is the UDP ping, which sends a UDP packet to the given ports. For most ports, the packet will be empty, though for a few a protocol-specific payload will be sent that is more likely to get a response.. The payload database is described at <http://nmap.org/book/nmap-payloads.html>.

The **--data-length** option can be used to send a fixed-length random payload to every port or (if you specify a value of 0) to disable payloads. You can also disable payloads by specifying **--data-length 0**.

The port list takes the same format as with the previously discussed **-PS** and **-PA** options. If no ports are specified, the default is 40125.. This default can be configured at compile-time by changing *DEFAULT_UDP_PROBE_PORT_SPEC*. in *nmap.h*.. A highly uncommon port is used by default because sending to open ports is often undesirable for this particular scan type.

Upon hitting a closed port on the target machine, the UDP probe should elicit an ICMP port unreachable packet in return. This signifies to Nmap that the machine is up and available. Many other types of ICMP errors, such as host/network unreachables or TTL exceeded are indicative of a down or unreachable host. A lack of response is also interpreted this way. If an open port is reached, most services simply ignore the empty packet and fail to return any response. This is why the default probe port is 40125, which is highly unlikely to be in use. A few services, such as the Character Generator (chargen) protocol, will respond to an empty UDP packet, and thus disclose to Nmap that the machine is available.

The primary advantage of this scan type is that it bypasses firewalls and filters that only screen TCP. For example, I once owned a Linksys BEFW11S4 wireless broadband router. The external interface of this device filtered all TCP ports by default, but UDP probes would still elicit port unreachable messages and thus give away the device.

-PY *port list* (SCTP INIT Ping) .

This option sends an SCTP packet containing a minimal INIT chunk. The default destination port is 80 (configurable at compile time by changing *DEFAULT_SCTP_PROBE_PORT_SPEC*. in *nmap.h*). Alternate ports can be specified as a parameter. The syntax is the same as for the **-p** except that port type specifiers like *S:* are not allowed. Examples are **-PY22** and **-PY22,80,179,5060**. Note that there can be no space between **-PY** and the port list. If multiple probes are specified they will be sent in parallel.

The INIT chunk suggests to the remote system that you are attempting to establish an association. Normally the destination port will be closed, and an ABORT chunk will be sent back. If the port happens to be open, the target will take the second step of an SCTP four-way-handshake. by responding with an INIT-ACK chunk. If the machine running Nmap has a functional SCTP stack, then it tears down the nascent association by responding with an ABORT chunk rather than sending a COOKIE-ECHO chunk which would be the next step in the four-way-handshake. The ABORT packet is sent by the kernel of the machine running Nmap in response to the unexpected INIT-ACK, not by Nmap itself.

Nmap does not care whether the port is open or closed. Either the ABORT or INIT-ACK response

discussed previously tell Nmap that the host is available and responsive.

On Unix boxes, only the privileged user root. is generally able to send and receive raw SCTP packets.. Using SCTP INIT Pings is currently not possible for unprivileged users..

-PE; -PP; -PM (ICMP Ping Types) .

In addition to the unusual TCP, UDP and SCTP host discovery types discussed previously, Nmap can send the standard packets sent by the ubiquitous ping program. Nmap sends an ICMP type 8 (echo request) packet to the target IP addresses, expecting a type 0 (echo reply) in return from available hosts.. Unfortunately for network explorers, many hosts and firewalls now block these packets, rather than responding as required by [RFC 1122](#)^[2].. For this reason, ICMP-only scans are rarely reliable enough against unknown targets over the Internet. But for system administrators monitoring an internal network, they can be a practical and efficient approach. Use the **-PE** option to enable this echo request behavior.

While echo request is the standard ICMP ping query, Nmap does not stop there. The ICMP standards ([RFC 792](#)^[3]. and [RFC 950](#)^[4]. “a host SHOULD NOT implement these messages”. Timestamp and address mask queries can be sent with the **-PP** and **-PM** options, respectively. A timestamp reply (ICMP code 14) or address mask reply (code 18) discloses that the host is available. These two queries can be valuable when administrators specifically block echo request packets while forgetting that other ICMP queries can be used for the same purpose.

-PO *protocol list* (IP Protocol Ping) .

One of the newer host discovery options is the IP protocol ping, which sends IP packets with the specified protocol number set in their IP header. The protocol list takes the same format as do port lists in the previously discussed TCP, UDP and SCTP host discovery options. If no protocols are specified, the default is to send multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4). The default protocols can be configured at compile-time by changing `DEFAULT_PROTO_PROBE_PORT_SPEC.` in `nmap.h`. Note that for the ICMP, IGMP, TCP (protocol 6), UDP (protocol 17) and SCTP (protocol 132), the packets are sent with the proper protocol headers. while other protocols are sent with no additional data beyond the IP header (unless the **---data-length.** option is specified).

This host discovery method looks for either responses using the same protocol as a probe, or ICMP protocol unreachable messages which signify that the given protocol isn't supported on the destination host. Either type of response signifies that the target host is alive.

-PR (ARP Ping) .

One of the most common Nmap usage scenarios is to scan an ethernet LAN. On most LANs, especially those using private address ranges specified by [RFC 1918](#)^[5], the vast majority of IP addresses are unused at any given time. When Nmap tries to send a raw IP packet such as an ICMP echo request, the operating system must determine the destination hardware (ARP) address corresponding to the target IP so that it can properly address the ethernet frame. This is often slow and problematic, since operating systems weren't written with the expectation that they would need to do millions of ARP requests against unavailable hosts in a short time period.

ARP scan puts Nmap and its optimized algorithms in charge of ARP requests. And if it gets a response back, Nmap doesn't even need to worry about the IP-based ping packets since it already knows the host is up. This makes ARP scan much faster and more reliable than IP-based scans. So it is done by default when scanning ethernet hosts that Nmap detects are on a local ethernet network. Even if different ping types (such as **-PE** or **-PS**) are specified, Nmap uses ARP instead for any of the targets which are on the same LAN. If you absolutely don't want to do an ARP scan, specify **---disable-arp-ping.**

For IPv6 (**-6** option), **-PR** uses ICMPv6 Neighbor Discovery instead of ARP. Neighbor Discovery, defined in RFC 4861, can be seen as the IPv6 equivalent of ARP.

---disable-arp-ping (No ARP or ND Ping) .

Nmap normally does ARP or IPv6 Neighbor Discovery (ND) discovery of locally connected ethernet hosts, even if other host discovery options such as **-Pn** or **-PE** are used. To disable this implicit behavior, use the **---disable-arp-ping** option.

The default behavior is normally faster, but this option is useful on networks using proxy ARP, in which a router speculatively replies to all ARP requests, making every target appear to be up according to ARP scan.

---traceroute (Trace path to host) .

Traceroutes are performed post-scan using information from the scan results to determine the port and protocol most likely to reach the target. It works with all scan types except connect scans (**-sT**) and idle scans (**-sI**). All traces use Nmap's dynamic timing model and are performed in parallel.

Traceroute works by sending packets with a low TTL (time-to-live) in an attempt to elicit ICMP Time Exceeded messages from intermediate hops between the scanner and the target host. Standard traceroute implementations start with a TTL of 1 and increment the TTL until the destination host is reached. Nmap's traceroute starts with a high TTL and then decrements the TTL until it reaches zero. Doing it backwards lets Nmap employ clever caching algorithms to speed up traces over multiple hosts. On average Nmap sends 5–10 fewer packets per host, depending on network conditions. If a single subnet is being scanned (i.e. 192.168.0.0/24) Nmap may only have to send two packets to most hosts.

-n (No DNS resolution) .

Tells Nmap to *never* do reverse DNS resolution on the active IP addresses it finds. Since DNS can be slow even with Nmap's built-in parallel stub resolver, this option can slash scanning times.

-R (DNS resolution for all targets) .

Tells Nmap to *always* do reverse DNS resolution on the target IP addresses. Normally reverse DNS is only performed against responsive (online) hosts.

---system-dns (Use system DNS resolver) .

By default, Nmap resolves IP addresses by sending queries directly to the name servers configured on your host and then listening for responses. Many requests (often dozens) are performed in parallel to improve performance. Specify this option to use your system resolver instead (one IP at a time via the **getnameinfo** call). This is slower and rarely useful unless you find a bug in the Nmap parallel resolver (please let us know if you do). The system resolver is always used for IPv6 scans.

---dns-servers *server1[,server2[,...]]* (Servers to use for reverse DNS queries) .

By default, Nmap determines your DNS servers (for rDNS resolution) from your **resolv.conf** file (Unix) or the Registry (Win32). Alternatively, you may use this option to specify alternate servers. This option is not honored if you are using **---system-dns** or an IPv6 scan. Using multiple DNS servers is often faster, especially if you choose authoritative servers for your target IP space. This option can also improve stealth, as your requests can be bounced off just about any recursive DNS server on the Internet.

This option also comes in handy when scanning private networks. Sometimes only a few name servers provide proper rDNS information, and you may not even know where they are. You can scan the network for port 53 (perhaps with version detection), then try Nmap list scans (**-sL**) specifying each name server one at a time with **---dns-servers** until you find one which works.

PORT SCANNING BASICS

While Nmap has grown in functionality over the years, it began as an efficient port scanner, and that remains its core function. The simple command **nmap target** scans 1,000 TCP ports on the host *target*. While many port scanners have traditionally lumped all ports into the open or closed states, Nmap is much more granular. It divides ports into six states: open, closed, filtered, unfiltered, open|filtered, or closed|filtered.

These states are not intrinsic properties of the port itself, but describe how Nmap sees them. For example,

an Nmap scan from the same network as the target may show port 135/tcp as open, while a scan at the same time with the same options from across the Internet might show that port as filtered.

The six port states recognized by Nmap

An application is actively accepting TCP connections, UDP datagrams or SCTP associations on this port. Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack. Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users. Open ports are also interesting for non-security scans because they show services available for use on the network.

A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it. They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up. Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next.

Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. The filtering could be from a dedicated firewall device, router rules, or host-based firewall software. These ports frustrate attackers because they provide so little information. Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited), but filters that simply drop probes without responding are far more common. This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering. This slows down the scan dramatically.

The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed. Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state. Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.

Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response. The lack of response could also mean that a packet filter dropped the probe or any response it elicited. So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way.

This state is used when Nmap is unable to determine whether a port is closed or filtered. It is only used for the IP ID idle scan.

PORT SCANNING TECHNIQUES

As a novice performing automotive repair, I can struggle for hours trying to fit my rudimentary tools (hammer, duct tape, wrench, etc.) to the task at hand. When I fail miserably and tow my jalopy to a real mechanic, he invariably fishes around in a huge tool chest until pulling out the perfect gizmo which makes the job seem effortless. The art of port scanning is similar. Experts understand the dozens of scan techniques and choose the appropriate one (or combination) for a given task. Inexperienced users and script kiddies, on the other hand, try to solve every problem with the default SYN scan. Since Nmap is free, the only barrier to port scanning mastery is knowledge. That certainly beats the automotive world, where it may take great skill to determine that you need a strut spring compressor, then you still have to pay thousands of dollars for it.

Most of the scan types are only available to privileged users.. This is because they send and receive raw packets, which requires root access on Unix systems. Using an administrator account on Windows is recommended, though Nmap sometimes works for unprivileged users on that platform when WinPcap has already been loaded into the OS. Requiring root privileges was a serious limitation when Nmap was released in 1997, as many users only had access to shared shell accounts. Now, the world is different. Computers are cheaper, far more people have always-on direct Internet access, and desktop Unix systems (including Linux and Mac OS X) are prevalent. A Windows version of Nmap is now available, allowing it to run on even more desktops. For all these reasons, users have less need to run Nmap from limited shared shell accounts. This is fortunate, as the privileged options make Nmap far more powerful and flexible.

While Nmap attempts to produce accurate results, keep in mind that all of its insights are based on packets returned by the target machines (or firewalls in front of them). Such hosts may be untrustworthy and send responses intended to confuse or mislead Nmap. Much more common are non-RFC-compliant hosts that do not respond as they should to Nmap probes. FIN, NULL, and Xmas scans are particularly susceptible to this problem. Such issues are specific to certain scan types and so are discussed in the individual scan type entries.

This section documents the dozen or so port scan techniques supported by Nmap. Only one method may be used at a time, except that UDP scan (**-sU**) and any one of the SCTP scan types (**-sY**, **-sZ**) may be combined with any one of the TCP scan types. As a memory aid, port scan type options are of the form **-sC**, where *C* is a prominent character in the scan name, usually the first. The one exception to this is the deprecated FTP bounce scan (**-b**). By default, Nmap performs a SYN Scan, though it substitutes a connect scan if the user does not have proper privileges to send raw packets (requires root access on Unix). Of the scans listed in this section, unprivileged users can only execute connect and FTP bounce scans.

-sS (TCP SYN scan) .

SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively unobtrusive and stealthy since it never completes TCP connections. SYN scan works against any compliant TCP stack rather than depending on idiosyncrasies of specific platforms as Nmap's FIN/NULL/Xmas, Maimon and idle scans do. It also allows clear, reliable differentiation between the open, closed, and filtered states.

This technique is often referred to as half-open scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and then wait for a response. A SYN/ACK indicates the port is listening (open), while a RST (reset) is indicative of a non-listener. If no response is received after several retransmissions, the port is marked as filtered. The port is also marked filtered if an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is received. The port is also considered open if a SYN packet (without the ACK flag) is received in response. This can be due to an extremely rare TCP feature known as a simultaneous open or split handshake connection (see <http://nmap.org/misc/split-handshake.pdf>).

-sT (TCP connect scan) .

TCP connect scan is the default TCP scan type when SYN scan is not an option. This is the case when a user does not have raw packet privileges. Instead of writing raw packets as most other scan types do, Nmap asks the underlying operating system to establish a connection with the target machine and port by issuing the **connect** system call. This is the same high-level system call that web browsers, P2P clients, and most other network-enabled applications use to establish a connection. It is part of a programming interface known as the Berkeley Sockets API. Rather than read raw packet responses off the wire, Nmap uses this API to obtain status information on each connection attempt.

When SYN scan is available, it is usually a better choice. Nmap has less control over the high level **connect** call than with raw packets, making it less efficient. The system call completes connections to open target ports rather than performing the half-open reset that SYN scan does. Not only does this take longer and require more packets to obtain the same information, but target machines are more likely to log the connection. A decent IDS will catch either, but most machines have no such alarm system. Many services on your average Unix system will add a note to syslog, and sometimes a cryptic error message, when Nmap connects and then closes the connection without sending data. Truly pathetic services crash when this happens, though that is uncommon. An administrator who sees a bunch of connection attempts in her logs from a single system should know that she has been connect scanned.

-sU (UDP scans) .

While most popular services on the Internet run over the TCP protocol, **UDP**^[6] services are widely deployed. DNS, SNMP, and DHCP (registered ports 53, 161/162, and 67/68) are three of the most common. Because UDP scanning is generally slower and more difficult than TCP, some security auditors ignore these ports. This is a mistake, as exploitable UDP services are quite common and

attackers certainly don't ignore the whole protocol. Fortunately, Nmap can help inventory UDP ports.

UDP scan is activated with the **-sU** option. It can be combined with a TCP scan type such as SYN scan (**-sS**) to check both protocols during the same run.

UDP scan works by sending a UDP packet to every targeted port. For some common ports such as 53 and 161, a protocol-specific payload is sent, but for most ports the packet is empty.. The **--data-length** option can be used to send a fixed-length random payload to every port or (if you specify a value of 0) to disable payloads. If an ICMP port unreachable error (type 3, code 3) is returned, the port is closed. Other ICMP unreachable errors (type 3, codes 1, 2, 9, 10, or 13) mark the port as filtered. Occasionally, a service will respond with a UDP packet, proving that it is open. If no response is received after retransmissions, the port is classified as open|filtered. This means that the port could be open, or perhaps packet filters are blocking the communication. Version detection (**-sV**) can be used to help differentiate the truly open ports from the filtered ones.

A big challenge with UDP scanning is doing it quickly. Open and filtered ports rarely send any response, leaving Nmap to time out and then conduct retransmissions just in case the probe or response were lost. Closed ports are often an even bigger problem. They usually send back an ICMP port unreachable error. But unlike the RST packets sent by closed TCP ports in response to a SYN or connect scan, many hosts rate limit ICMP port unreachable messages by default. Linux and Solaris are particularly strict about this. For example, the Linux 2.4.20 kernel limits destination unreachable messages to one per second (in net/ipv4/icmp.c).

Nmap detects rate limiting and slows down accordingly to avoid flooding the network with useless packets that the target machine will drop. Unfortunately, a Linux-style limit of one packet per second makes a 65,536-port scan take more than 18 hours. Ideas for speeding your UDP scans up include scanning more hosts in parallel, doing a quick scan of just the popular ports first, scanning from behind the firewall, and using **--host-timeout** to skip slow hosts.

-sY (SCTP INIT scan) .

SCTP^[7] is a relatively new alternative to the TCP and UDP protocols, combining most characteristics of TCP and UDP, and also adding new features like multi-homing and multi-streaming. It is mostly being used for SS7/SIGTRAN related services but has the potential to be used for other applications as well. SCTP INIT scan is the SCTP equivalent of a TCP SYN scan. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. Like SYN scan, INIT scan is relatively unobtrusive and stealthy, since it never completes SCTP associations. It also allows clear, reliable differentiation between the open, closed, and filtered states.

This technique is often referred to as half-open scanning, because you don't open a full SCTP association. You send an INIT chunk, as if you are going to open a real association and then wait for a response. An INIT-ACK chunk indicates the port is listening (open), while an ABORT chunk is indicative of a non-listener. If no response is received after several retransmissions, the port is marked as filtered. The port is also marked filtered if an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is received.

-sN; -sF; -sX (TCP NULL, FIN, and Xmas scans) .

These three scan types (even more are possible with the **--scanflags** option described in the next section) exploit a subtle loophole in the **TCP RFC**^[8] to differentiate between open and closed ports. Page 65 of RFC 793 says that "if the [destination] port state is CLOSED an incoming segment not containing a RST causes a RST to be sent in response." Then the next page discusses packets sent to open ports without the SYN, RST, or ACK bits set, stating that: "you are unlikely to get here, but if you do, drop the segment, and return."

When scanning systems compliant with this RFC text, any packet not containing SYN, RST, or ACK bits will result in a returned RST if the port is closed and no response at all if the port is open. As long

as none of those three bits are included, any combination of the other three (FIN, PSH, and URG) are OK. Nmap exploits this with three scan types:

Null scan (**-sN**)

Does not set any bits (TCP flag header is 0)

FIN scan (**-sF**)

Sets just the TCP FIN bit.

Xmas scan (**-sX**)

Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.

These three scan types are exactly the same in behavior except for the TCP flags set in probe packets. If a RST packet is received, the port is considered closed, while no response means it is open|filtered. The port is marked filtered if an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is received.

The key advantage to these scan types is that they can sneak through certain non-stateful firewalls and packet filtering routers. Another advantage is that these scan types are a little more stealthy than even a SYN scan. Don't count on this though—most modern IDS products can be configured to detect them. The big downside is that not all systems follow RFC 793 to the letter. A number of systems send RST responses to the probes regardless of whether the port is open or not. This causes all of the ports to be labeled closed. Major operating systems that do this are Microsoft Windows, many Cisco devices, BSDI, and IBM OS/400. This scan does work against most Unix-based systems though. Another downside of these scans is that they can't distinguish open ports from certain filtered ones, leaving you with the response open|filtered.

-sA (TCP ACK scan) .

This scan is different than the others discussed so far in that it never determines open (or even open|filtered) ports. It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.

The ACK scan probe packet has only the ACK flag set (unless you use **--scanflags**). When scanning unfiltered systems, open and closed ports will both return a RST packet. Nmap then labels them as unfiltered, meaning that they are reachable by the ACK packet, but whether they are open or closed is undetermined. Ports that don't respond, or send certain ICMP error messages back (type 3, code 1, 2, 3, 9, 10, or 13), are labeled filtered.

-sW (TCP Window scan) .

Window scan is exactly the same as ACK scan except that it exploits an implementation detail of certain systems to differentiate open ports from closed ones, rather than always printing unfiltered when a RST is returned. It does this by examining the TCP Window field of the RST packets returned. On some systems, open ports use a positive window size (even for RST packets) while closed ones have a zero window. So instead of always listing a port as unfiltered when it receives a RST back, Window scan lists the port as open or closed if the TCP Window value in that reset is positive or zero, respectively.

This scan relies on an implementation detail of a minority of systems out on the Internet, so you can't always trust it. Systems that don't support it will usually return all ports closed. Of course, it is possible that the machine really has no open ports. If most scanned ports are closed but a few common port numbers (such as 22, 25, 53) are filtered, the system is most likely susceptible. Occasionally, systems will even show the exact opposite behavior. If your scan shows 1,000 open ports and three closed or filtered ports, then those three may very well be the truly open ones.

-sM (TCP Maimon scan) .

The Maimon scan is named after its discoverer, Uriel Maimon.. He described the technique in Phrack Magazine issue #49 (November 1996).. Nmap, which included this technique, was released two issues later. This technique is exactly the same as NULL, FIN, and Xmas scans, except that the probe is FIN/ACK. According to [RFC 793](#)^[8] (TCP), a RST packet should be generated in response to such a

probe whether the port is open or closed. However, Uriel noticed that many BSD-derived systems simply drop the packet if the port is open.

--scanflags (Custom TCP scan) .

Truly advanced Nmap users need not limit themselves to the canned scan types offered. The **--scanflags** option allows you to design your own scan by specifying arbitrary TCP flags.. Let your creative juices flow, while evading intrusion detection systems. whose vendors simply paged through the Nmap man page adding specific rules!

The **--scanflags** argument can be a numerical flag value such as 9 (PSH and FIN), but using symbolic names is easier. Just mash together any combination of URG, ACK, PSH, RST, SYN, and FIN. For example, **--scanflags URGACKPSHRSTSYNFIN** sets everything, though it's not very useful for scanning. The order these are specified in is irrelevant.

In addition to specifying the desired flags, you can specify a TCP scan type (such as **-sA** or **-sF**). That base type tells Nmap how to interpret responses. For example, a SYN scan considers no-response to indicate a filtered port, while a FIN scan treats the same as open|filtered. Nmap will behave the same way it does for the base scan type, except that it will use the TCP flags you specify instead. If you don't specify a base type, SYN scan is used.

-sZ (SCTP COOKIE ECHO scan) .

SCTP COOKIE ECHO scan is a more advanced SCTP scan. It takes advantage of the fact that SCTP implementations should silently drop packets containing COOKIE ECHO chunks on open ports, but send an ABORT if the port is closed. The advantage of this scan type is that it is not as obvious a port scan than an INIT scan. Also, there may be non-stateful firewall rulesets blocking INIT chunks, but not COOKIE ECHO chunks. Don't be fooled into thinking that this will make a port scan invisible; a good IDS will be able to detect SCTP COOKIE ECHO scans too. The downside is that SCTP COOKIE ECHO scans cannot differentiate between open and filtered ports, leaving you with the state open|filtered in both cases.

-sI *zombie host[:probeport]* (idle scan) .

This advanced scan method allows for a truly blind TCP port scan of the target (meaning no packets are sent to the target from your real IP address). Instead, a unique side-channel attack exploits predictable IP fragmentation ID sequence generation on the zombie host to glean information about the open ports on the target. IDS systems will display the scan as coming from the zombie machine you specify (which must be up and meet certain criteria). This fascinating scan type is too complex to fully describe in this reference guide, so I wrote and posted an informal paper with full details at <http://nmap.org/book/idlescan.html>.

Besides being extraordinarily stealthy (due to its blind nature), this scan type permits mapping out IP-based trust relationships between machines. The port listing shows open ports *from the perspective of the zombie host*. So you can try scanning a target using various zombies that you think might be trusted. (via router/packet filter rules).

You can add a colon followed by a port number to the zombie host if you wish to probe a particular port on the zombie for IP ID changes. Otherwise Nmap will use the port it uses by default for TCP pings (80).

-sO (IP protocol scan) .

IP protocol scan allows you to determine which IP protocols (TCP, ICMP, IGMP, etc.) are supported by target machines. This isn't technically a port scan, since it cycles through IP protocol numbers rather than TCP or UDP port numbers. Yet it still uses the **-p** option to select scanned protocol numbers, reports its results within the normal port table format, and even uses the same underlying scan engine as the true port scanning methods. So it is close enough to a port scan that it belongs here.

Besides being useful in its own right, protocol scan demonstrates the power of open-source software. While the fundamental idea is pretty simple, I had not thought to add it nor received any requests for

such functionality. Then in the summer of 2000, Gerhard Rieger conceived the idea, wrote an excellent patch implementing it, and sent it to the nmap-hackers mailing list. I incorporated that patch into the Nmap tree and released a new version the next day. Few pieces of commercial software have users enthusiastic enough to design and contribute their own improvements!

Protocol scan works in a similar fashion to UDP scan. Instead of iterating through the port number field of a UDP packet, it sends IP packet headers and iterates through the eight-bit IP protocol field. The headers are usually empty, containing no data and not even the proper header for the claimed protocol. The exceptions are TCP, UDP, ICMP, SCTP, and IGMP. A proper protocol header for those is included since some systems won't send them otherwise and because Nmap already has functions to create them. Instead of watching for ICMP port unreachable messages, protocol scan is on the lookout for ICMP *protocol* unreachable messages. If Nmap receives any response in any protocol from the target host, Nmap marks that protocol as open. An ICMP protocol unreachable error (type 3, code 2) causes the protocol to be marked as closed. Other ICMP unreachable errors (type 3, code 1, 3, 9, 10, or 13) cause the protocol to be marked filtered (though they prove that ICMP is open at the same time). If no response is received after retransmissions, the protocol is marked open|filtered.

-b *FTP relay host* (FTP bounce scan) .

An interesting feature of the FTP protocol ([RFC 959](#)^[9]) is support for so-called proxy FTP connections. This allows a user to connect to one FTP server, then ask that files be sent to a third-party server. Such a feature is ripe for abuse on many levels, so most servers have ceased supporting it. One of the abuses this feature allows is causing the FTP server to port scan other hosts. Simply ask the FTP server to send a file to each interesting port of a target host in turn. The error message will describe whether the port is open or not. This is a good way to bypass firewalls because organizational FTP servers are often placed where they have more access to other internal hosts than any old Internet host would. Nmap supports FTP bounce scan with the **-b** option. It takes an argument of the form *username:password@server:port*. *Server* is the name or IP address of a vulnerable FTP server. As with a normal URL, you may omit *username:password*, in which case anonymous login credentials (user: anonymous password:-wwwuser@) are used. The port number (and preceding colon) may be omitted as well, in which case the default FTP port (21) on *server* is used.

This vulnerability was widespread in 1997 when Nmap was released, but has largely been fixed. Vulnerable servers are still around, so it is worth trying when all else fails. If bypassing a firewall is your goal, scan the target network for port 21 (or even for any FTP services if you scan all ports with version detection) and use the ftp-bounce. NSE script. Nmap will tell you whether the host is vulnerable or not. If you are just trying to cover your tracks, you don't need to (and, in fact, shouldn't) limit yourself to hosts on the target network. Before you go scanning random Internet addresses for vulnerable FTP servers, consider that sysadmins may not appreciate you abusing their servers in this way.

PORT SPECIFICATION AND SCAN ORDER

In addition to all of the scan methods discussed previously, Nmap offers options for specifying which ports are scanned and whether the scan order is randomized or sequential. By default, Nmap scans the most common 1,000 ports for each protocol.

-p *port ranges* (Only scan specified ports) .

This option specifies which ports you want to scan and overrides the default. Individual port numbers are OK, as are ranges separated by a hyphen (e.g. 1-1023). The beginning and/or end values of a range may be omitted, causing Nmap to use 1 and 65535, respectively. So you can specify **-p-** to scan ports from 1 through 65535. Scanning port zero. is allowed if you specify it explicitly. For IP protocol scanning (**-sO**), this option specifies the protocol numbers you wish to scan for (0-255).

When scanning a combination of protocols (e.g. TCP and UDP), you can specify a particular protocol by preceding the port numbers by T: for TCP, U: for UDP, S: for SCTP, or P: for IP Protocol. The qualifier lasts until you specify another qualifier. For example, the argument **-p U:53,111,137,T:21-25,80,139,8080** would scan UDP ports 53, 111, and 137, as well as the listed TCP

ports. Note that to scan both UDP and TCP, you have to specify **-sU** and at least one TCP scan type (such as **-sS**, **-sF**, or **-sT**). If no protocol qualifier is given, the port numbers are added to all protocol lists. Ports can also be specified by name according to what the port is referred to in the `nmap-services`. You can even use the wildcards `*` and `?` with the names. For example, to scan FTP and all ports whose names begin with "http", use **-p ftp,http***. Be careful about shell expansions and quote the argument to **-p** if unsure.

Ranges of ports can be surrounded by square brackets to indicate ports inside that range that appear in `nmap-services`. For example, the following will scan all ports in `nmap-services` equal to or below 1024: **-p [-1024]**. Be careful with shell expansions and quote the argument to **-p** if unsure.

-F (Fast (limited port) scan) .

Specifies that you wish to scan fewer ports than the default. Normally Nmap scans the most common 1,000 ports for each scanned protocol. With **-F**, this is reduced to 100.

Nmap needs an `nmap-services` file with frequency information in order to know which ports are the most common. If port frequency information isn't available, perhaps because of the use of a custom `nmap-services` file, Nmap scans all named ports plus ports 1–1024. In that case, **-F** means to scan only ports that are named in the services file.

-r (Don't randomize ports) .

By default, Nmap randomizes the scanned port order (except that certain commonly accessible ports are moved near the beginning for efficiency reasons). This randomization is normally desirable, but you can specify **-r** for sequential (sorted from lowest to highest) port scanning instead.

--port-ratio *ratio*<decimal number between 0 and 1>

Scans all ports in `nmap-services` file with a ratio greater than the one given. *ratio* must be between 0.0 and 1.1.

--top-ports *n*

Scans the *n* highest-ratio ports found in `nmap-services` file. *n* must be 1 or greater.

SERVICE AND VERSION DETECTION

Point Nmap at a remote machine and it might tell you that ports 25/tcp, 80/tcp, and 53/udp are open. Using its `nmap-services` database of about 2,200 well-known services, Nmap would report that those ports probably correspond to a mail server (SMTP), web server (HTTP), and name server (DNS) respectively. This lookup is usually accurate—the vast majority of daemons listening on TCP port 25 are, in fact, mail servers. However, you should not bet your security on this! People can and do run services on strange ports.

Even if Nmap is right, and the hypothetical server above is running SMTP, HTTP, and DNS servers, that is not a lot of information. When doing vulnerability assessments (or even simple network inventories) of your companies or clients, you really want to know which mail and DNS servers and versions are running. Having an accurate version number helps dramatically in determining which exploits a server is vulnerable to. Version detection helps you obtain this information.

After TCP and/or UDP ports are discovered using one of the other scan methods, version detection interrogates those ports to determine more about what is actually running. The `nmap-service-probes` database contains probes for querying various services and match expressions to recognize and parse responses. Nmap tries to determine the service protocol (e.g. FTP, SSH, Telnet, HTTP), the application name (e.g. ISC BIND, Apache httpd, Solaris telnetd), the version number, hostname, device type (e.g. printer, router), the OS family (e.g. Windows, Linux). When possible, Nmap also gets the Common Platform Enumeration (CPE) representation of this information. Sometimes miscellaneous details like whether an X server is open to connections, the SSH protocol version, or the KaZaA user name, are available. Of course, most services don't provide all of this information. If Nmap was compiled with OpenSSL support, it will connect to SSL servers to deduce the service listening behind that encryption layer. Some UDP ports are left in the open|filtered state after a UDP port scan is unable to determine whether the port is open or filtered. Version detection will try to elicit a response from these ports (just as it does with open ports), and change the state to open if it succeeds. open|filtered TCP ports are treated the

same way. Note that the Nmap **-A** option enables version detection among other things. A paper documenting the workings, usage, and customization of version detection is available at <http://nmap.org/book/vscan.html>.

When RPC services are discovered, the Nmap RPC grinder. is automatically used to determine the RPC program and version numbers. It takes all the TCP/UDP ports detected as RPC and floods them with SunRPC program NULL commands in an attempt to determine whether they are RPC ports, and if so, what program and version number they serve up. Thus you can effectively obtain the same info as **rpcinfo -p** even if the target's portmapper is behind a firewall (or protected by TCP wrappers). Decoys do not currently work with RPC scan..

When Nmap receives responses from a service but cannot match them to its database, it prints out a special fingerprint and a URL for you to submit if to if you know for sure what is running on the port. Please take a couple minutes to make the submission so that your find can benefit everyone. Thanks to these submissions, Nmap has about 6,500 pattern matches for more than 650 protocols such as SMTP, FTP, HTTP, etc..

Version detection is enabled and controlled with the following options:

-sV (Version detection) .

Enables version detection, as discussed above. Alternatively, you can use **-A**, which enables version detection among other things.

-sR. is an alias for **-sV**. Prior to March 2011, it was used to active the RPC grinder separately from version detection, but now these options are always combined.

--allports (Don't exclude any ports from version detection) .

By default, Nmap version detection skips TCP port 9100 because some printers simply print anything sent to that port, leading to dozens of pages of HTTP GET requests, binary SSL session requests, etc. This behavior can be changed by modifying or removing the Exclude directive in `nmap-service-probes`, or you can specify **--allports** to scan all ports regardless of any Exclude directive.

--version-intensity *intensity* (Set version scan intensity) .

When performing a version scan (**-sV**), Nmap sends a series of probes, each of which is assigned a rarity value between one and nine. The lower-numbered probes are effective against a wide variety of common services, while the higher-numbered ones are rarely useful. The intensity level specifies which probes should be applied. The higher the number, the more likely it is the service will be correctly identified. However, high intensity scans take longer. The intensity must be between 0 and 9.. The default is 7.. When a probe is registered to the target port via the `nmap-service-probes` ports directive, that probe is tried regardless of intensity level. This ensures that the DNS probes will always be attempted against any open port 53, the SSL probe will be done against 443, etc.

--version-light (Enable light mode) .

This is a convenience alias for **--version-intensity 2**. This light mode makes version scanning much faster, but it is slightly less likely to identify services.

--version-all (Try every single probe) .

An alias for **--version-intensity 9**, ensuring that every single probe is attempted against each port.

--version-trace (Trace version scan activity) .

This causes Nmap to print out extensive debugging info about what version scanning is doing. It is a subset of what you get with **--packet-trace**.

OS DETECTION

One of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting. Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After performing dozens of tests such as TCP ISN sampling, TCP options support and ordering, IP ID sampling, and the initial window size check, Nmap compares the results to its `nmap-os-db`. database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match. Each fingerprint includes a freeform textual description of the OS, and a classification which provides the vendor name (e.g.

Sun), underlying OS (e.g. Solaris), OS generation (e.g. 10), and device type (general purpose, router, switch, game console, etc). Most fingerprints also have a Common Platform Enumeration (CPE) representation, like `cpe:/o:linux:linux_kernel:2.6`.

If Nmap is unable to guess the OS of a machine, and conditions are good (e.g. at least one open port and one closed port were found), Nmap will provide a URL you can use to submit the fingerprint if you know (for sure) the OS running on the machine. By doing this you contribute to the pool of operating systems known to Nmap and thus it will be more accurate for everyone.

OS detection enables some other tests which make use of information that is gathered during the process anyway. One of these is TCP Sequence Predictability Classification. This measures approximately how hard it is to establish a forged TCP connection against the remote host. It is useful for exploiting source-IP based trust relationships (rlogin, firewall filters, etc) or for hiding the source of an attack. This sort of spoofing is rarely performed any more, but many machines are still vulnerable to it. The actual difficulty number is based on statistical sampling and may fluctuate. It is generally better to use the English classification such as “worthy challenge” or “trivial joke”. This is only reported in normal output in verbose (`-v`) mode. When verbose mode is enabled along with `-O`, IP ID sequence generation is also reported. Most machines are in the “incremental” class, which means that they increment the ID field in the IP header for each packet they send. This makes them vulnerable to several advanced information gathering and spoofing attacks.

Another bit of extra information enabled by OS detection is a guess at a target's uptime. This uses the TCP timestamp option ([RFC 1323](#)^[10]) to guess when a machine was last rebooted. The guess can be inaccurate due to the timestamp counter not being initialized to zero or the counter overflowing and wrapping around, so it is printed only in verbose mode.

A paper documenting the workings, usage, and customization of OS detection is available at <http://nmap.org/book/osdetect.html>.

OS detection is enabled and controlled with the following options:

-O (Enable OS detection) .

Enables OS detection, as discussed above. Alternatively, you can use `-A` to enable OS detection along with other things.

--osscan-limit (Limit OS detection to promising targets) .

OS detection is far more effective if at least one open and one closed TCP port are found. Set this option and Nmap will not even try OS detection against hosts that do not meet this criteria. This can save substantial time, particularly on `-Pn` scans against many hosts. It only matters when OS detection is requested with `-O` or `-A`.

--osscan-guess; --fuzzy (Guess OS detection results) .

When Nmap is unable to detect a perfect OS match, it sometimes offers up near-matches as possibilities. The match has to be very close for Nmap to do this by default. Either of these (equivalent) options make Nmap guess more aggressively. Nmap will still tell you when an imperfect match is printed and display its confidence level (percentage) for each guess.

--max-os-tries (Set the maximum number of OS detection tries against a target) .

When Nmap performs OS detection against a target and fails to find a perfect match, it usually repeats the attempt. By default, Nmap tries five times if conditions are favorable for OS fingerprint submission, and twice when conditions aren't so good. Specifying a lower **--max-os-tries** value (such as 1) speeds Nmap up, though you miss out on retries which could potentially identify the OS. Alternatively, a high value may be set to allow even more retries when conditions are favorable. This is rarely done, except to generate better fingerprints for submission and integration into the Nmap OS database.

NMAP SCRIPTING ENGINE (NSE)

The Nmap Scripting Engine (NSE) is one of Nmap's most powerful and flexible features. It allows users to write (and share) simple scripts (using the [Lua programming language](#)^[11],

Tasks we had in mind when creating the system include network discovery, more sophisticated version

detection, vulnerability detection. NSE can even be used for vulnerability exploitation.

To reflect those different uses and to simplify the choice of which scripts to run, each script contains a field associating it with one or more categories. Currently defined categories are auth, broadcast, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, and vuln. These are all described at <http://nmap.org/book/nse-usage.html#nse-categories>.

Scripts are not run in a sandbox and thus could accidentally or maliciously damage your system or invade your privacy. Never run scripts from third parties unless you trust the authors or have carefully audited the scripts yourself.

The Nmap Scripting Engine is described in detail at <http://nmap.org/book/nse.html>

and is controlled by the following options:

-sC .

Performs a script scan using the default set of scripts. It is equivalent to **--script=default**. Some of the scripts in this category are considered intrusive and should not be run against a target network without permission.

--script filename|category|directory|expression[,...] .

Runs a script scan using the comma-separated list of filenames, script categories, and directories. Each element in the list may also be a Boolean expression describing a more complex set of scripts. Each element is interpreted first as an expression, then as a category, and finally as a file or directory name.

There are two special features for advanced users only. One is to prefix script names and expressions with + to force them to run even if they normally wouldn't (e.g. the relevant service wasn't detected on the target port). The other is that the argument all may be used to specify every script in Nmap's database. Be cautious with this because NSE contains dangerous scripts such as exploits, brute force authentication crackers, and denial of service attacks.

File and directory names may be relative or absolute. Absolute names are used directly. Relative paths are looked for in the scripts of each of the following places until found:

--datadir

\$NMAPDIR.

`~/nmap` (not searched on Windows).

`HOME\AppData\Roaming\nmap` (only on Windows).

the directory containing the nmap executable

the directory containing the nmap executable, followed by `./share/nmap`

NMAPDATADIR.

the current directory.

When a directory name is given, Nmap loads every file in the directory whose name ends with `.nse`. All other files are ignored and directories are not searched recursively. When a filename is given, it does not have to have the `.nse` extension; it will be added automatically if necessary. Nmap scripts are stored in a scripts subdirectory of the Nmap data directory by default (see <http://nmap.org/book/data-files.html>).

For efficiency, scripts are indexed in a database stored in `scripts/script.db.` which lists the category or categories in which each script belongs. When referring to scripts from `script.db` by name, you can use a shell-style `*` wildcard.

nmap --script "http-*

Loads all scripts whose name starts with `http-`, such as `http-auth` and `http-open-proxy`. The argument to **--script** had to be in quotes to protect the wildcard from the shell.

More complicated script selection can be done using the and, or, and not operators to build Boolean

expressions. The operators have the same [precedence](#)^[12] as in Lua: not is the highest, followed by and and then or. You can alter precedence by using parentheses. Because expressions contain space characters it is necessary to quote them.

nmap --script "not intrusive"

Loads every script except for those in the intrusive category.

nmap --script "default or safe"

This is functionally equivalent to **nmap --script "default,safe"**. It loads all scripts that are in the default category or the safe category or both.

nmap --script "default and safe"

Loads those scripts that are in *both* the default and safe categories.

nmap --script "(default or safe or intrusive) and not http-"

Loads scripts in the default, safe, or intrusive categories, except for those whose names start with http-.

--script-args n1=v1,n2={n3=v3},n4={v4,v5} .

Lets you provide arguments to NSE scripts. Arguments are a comma-separated list of name=value pairs. Names and values may be strings not containing whitespace or the characters '{', '}', '=', or ';'. To include one of these characters in a string, enclose the string in single or double quotes. Within a quoted string, '\ ' escapes a quote. A backslash is only used to escape quotation marks in this special case; in all other cases a backslash is interpreted literally. Values may also be tables enclosed in {}, just as in Lua. A table may contain simple string values or more name-value pairs, including nested tables. Many scripts qualify their arguments with the script name, as in xmpp-info.server_name. You may use that full qualified version to affect just the specified script, or you may pass the unqualified version (server_name in this case) to affect all scripts using that argument name. A script will first check for its fully qualified argument name (the name specified in its documentation) before it accepts an unqualified argument name. A complex example of script arguments is **--script-args 'user=foo,pass="{ }=bar",whois={whodb=nofollow+ripe},xmpp-info.server_name=localhost'**. The online NSE Documentation Portal at <http://nmap.org/nsedoc/> lists the arguments that each script accepts.

--script-args-file filename .

Lets you load arguments to NSE scripts from a file. Any arguments on the command line supersede ones in the file. The file can be an absolute path, or a path relative to Nmap's usual search path (NMAPDIR, etc.) Arguments can be comma-separated or newline-separated, but otherwise follow the same rules as for **--script-args**, without requiring special quoting and escaping, since they are not parsed by the shell.

--script-help filename|category|directory|expression|all[,...] .

Shows help about scripts. For each script matching the given specification, Nmap prints the script name, its categories, and its description. The specifications are the same as those accepted by **--script**; so for example if you want help about the ftp-anon script, you would run **nmap --script-help ftp-anon**. In addition to getting help for individual scripts, you can use this as a preview of what scripts will be run for a specification, for example with **nmap --script-help default**.

--script-trace .

This option does what **--packet-trace** does, just one ISO layer higher. If this option is specified all incoming and outgoing communication performed by a script is printed. The displayed information includes the communication protocol, the source, the target and the transmitted data. If more than 5% of all transmitted data is not printable, then the trace output is in a hex dump format. Specifying **--packet-trace** enables script tracing too.

--script-updatedb .

This option updates the script database found in scripts/script.db which is used by Nmap to determine the available default scripts and categories. It is only necessary to update the database if you have added or removed NSE scripts from the default scripts directory or if you have changed the categories of any script. This option is generally used by itself: **nmap --script-updatedb**.

TIMING AND PERFORMANCE

One of my highest Nmap development priorities has always been performance. A default scan (**nmap hostname**) of a host on my local network takes a fifth of a second. That is barely enough time to blink, but adds up when you are scanning hundreds or thousands of hosts. Moreover, certain scan options such as UDP scanning and version detection can increase scan times substantially. So can certain firewall configurations, particularly response rate limiting. While Nmap utilizes parallelism and many advanced algorithms to accelerate these scans, the user has ultimate control over how Nmap runs. Expert users carefully craft Nmap commands to obtain only the information they care about while meeting their time constraints.

Techniques for improving scan times include omitting non-critical tests, and upgrading to the latest version of Nmap (performance enhancements are made frequently). Optimizing timing parameters can also make a substantial difference. Those options are listed below.

Some options accept a time parameter. This is specified in seconds by default, though you can append 'ms', 's', 'm', or 'h' to the value to specify milliseconds, seconds, minutes, or hours. So the **--host-timeout** arguments 900000ms, 900, 900s, and 15m all do the same thing.

--min-hostgroup numhosts; **--max-hostgroup numhosts** (Adjust parallel scan group sizes) .

Nmap has the ability to port scan or version scan multiple hosts in parallel. Nmap does this by dividing the target IP space into groups and then scanning one group at a time. In general, larger groups are more efficient. The downside is that host results can't be provided until the whole group is finished. So if Nmap started out with a group size of 50, the user would not receive any reports (except for the updates offered in verbose mode) until the first 50 hosts are completed.

By default, Nmap takes a compromise approach to this conflict. It starts out with a group size as low as five so the first results come quickly and then increases the groupsize to as high as 1024. The exact default numbers depend on the options given. For efficiency reasons, Nmap uses larger group sizes for UDP or few-port TCP scans.

When a maximum group size is specified with **--max-hostgroup**, Nmap will never exceed that size. Specify a minimum size with **--min-hostgroup** and Nmap will try to keep group sizes above that level. Nmap may have to use smaller groups than you specify if there are not enough target hosts left on a given interface to fulfill the specified minimum. Both may be set to keep the group size within a specific range, though this is rarely desired.

These options do not have an effect during the host discovery phase of a scan. This includes plain ping scans (**-sn**). Host discovery always works in large groups of hosts to improve speed and accuracy.

The primary use of these options is to specify a large minimum group size so that the full scan runs more quickly. A common choice is 256 to scan a network in Class C sized chunks. For a scan with many ports, exceeding that number is unlikely to help much. For scans of just a few port numbers, host group sizes of 2048 or more may be helpful.

--min-parallelism numprobes; **--max-parallelism numprobes** (Adjust probe parallelization) .

These options control the total number of probes that may be outstanding for a host group. They are used for port scanning and host discovery. By default, Nmap calculates an ever-changing ideal parallelism based on network performance. If packets are being dropped, Nmap slows down and allows fewer outstanding probes. The ideal probe number slowly rises as the network proves itself worthy. These options place minimum or maximum bounds on that variable. By default, the ideal parallelism can drop to one if the network proves unreliable and rise to several hundred in perfect conditions.

The most common usage is to set **--min-parallelism** to a number higher than one to speed up scans of poorly performing hosts or networks. This is a risky option to play with, as setting it too high may affect accuracy. Setting this also reduces Nmap's ability to control parallelism dynamically based on network conditions. A value of 10 might be reasonable, though I only adjust this value as a last resort.

The **--max-parallelism** option is sometimes set to one to prevent Nmap from sending more than one probe at a time to hosts. The **--scan-delay** option, discussed later, is another way to do this.

--min-rtt-timeout *time*, **--max-rtt-timeout** *time*, **--initial-rtt-timeout** *time* (Adjust probe timeouts)

Nmap maintains a running timeout value for determining how long it will wait for a probe response before giving up or retransmitting the probe. This is calculated based on the response times of previous probes.

If the network latency shows itself to be significant and variable, this timeout can grow to several seconds. It also starts at a conservative (high) level and may stay that way for a while when Nmap scans unresponsive hosts.

Specifying a lower **--max-rtt-timeout** and **--initial-rtt-timeout** than the defaults can cut scan times significantly. This is particularly true for pingless (**-Pn**) scans, and those against heavily filtered networks. Don't get too aggressive though. The scan can end up taking longer if you specify such a low value that many probes are timing out and retransmitting while the response is in transit.

If all the hosts are on a local network, 100 milliseconds (**--max-rtt-timeout 100ms**) is a reasonable aggressive value. If routing is involved, ping a host on the network first with the ICMP ping utility, or with a custom packet crafter such as Nping. That is more likely to get through a firewall. Look at the maximum round trip time out of ten packets or so. You might want to double that for the **--initial-rtt-timeout** and triple or quadruple it for the **--max-rtt-timeout**. I generally do not set the maximum RTT below 100 ms, no matter what the ping times are. Nor do I exceed 1000 ms.

--min-rtt-timeout is a rarely used option that could be useful when a network is so unreliable that even Nmap's default is too aggressive. Since Nmap only reduces the timeout down to the minimum when the network seems to be reliable, this need is unusual and should be reported as a bug to the nmap-dev mailing list.

--max-retries *numtries* (Specify the maximum number of port scan probe retransmissions) .

When Nmap receives no response to a port scan probe, it could mean the port is filtered. Or maybe the probe or response was simply lost on the network. It is also possible that the target host has rate limiting enabled that temporarily blocked the response. So Nmap tries again by retransmitting the initial probe. If Nmap detects poor network reliability, it may try many more times before giving up on a port. While this benefits accuracy, it also lengthens scan times. When performance is critical, scans may be sped up by limiting the number of retransmissions allowed. You can even specify **--max-retries 0** to prevent any retransmissions, though that is only recommended for situations such as informal surveys where occasional missed ports and hosts are acceptable.

The default (with no **-T** template) is to allow ten retransmissions. If a network seems reliable and the target hosts aren't rate limiting, Nmap usually only does one retransmission. So most target scans aren't even affected by dropping **--max-retries** to a low value such as three. Such values can substantially speed scans of slow (rate limited) hosts. You usually lose some information when Nmap gives up on ports early, though that may be preferable to letting the **--host-timeout** expire and losing all information about the target.

--host-timeout *time* (Give up on slow target hosts) .

Some hosts simply take a *long* time to scan. This may be due to poorly performing or unreliable networking hardware or software, packet rate limiting, or a restrictive firewall. The slowest few percent of the scanned hosts can eat up a majority of the scan time. Sometimes it is best to cut your losses and skip those hosts initially. Specify **--host-timeout** with the maximum amount of time you are willing to wait. For example, specify 30m to ensure that Nmap doesn't waste more than half an hour on a single host. Note that Nmap may be scanning other hosts at the same time during that half an hour, so it isn't a complete loss. A host that times out is skipped. No port table, OS detection, or version detection results are printed for that host.

--scan-delay *time*; **--max-scan-delay** *time* (Adjust delay between probes) .

This option causes Nmap to wait at least the given amount of time between each probe it sends to a given host. This is particularly useful in the case of rate limiting.. Solaris machines (among many others) will usually respond to UDP scan probe packets with only one ICMP message per second. Any more than that sent by Nmap will be wasteful. A **--scan-delay** of 1s will keep Nmap at that slow rate. Nmap tries to detect rate limiting and adjust the scan delay accordingly, but it doesn't hurt to specify it explicitly if you already know what rate works best.

When Nmap adjusts the scan delay upward to cope with rate limiting, the scan slows down dramatically. The **--max-scan-delay** option specifies the largest delay that Nmap will allow. A low **--max-scan-delay** can speed up Nmap, but it is risky. Setting this value too low can lead to wasteful packet retransmissions and possible missed ports when the target implements strict rate limiting.

Another use of **--scan-delay** is to evade threshold based intrusion detection and prevention systems (IDS/IPS)..

--min-rate *number*; **--max-rate** *number* (Directly control the scanning rate) .

Nmap's dynamic timing does a good job of finding an appropriate speed at which to scan. Sometimes, however, you may happen to know an appropriate scanning rate for a network, or you may have to guarantee that a scan will be finished by a certain time. Or perhaps you must keep Nmap from scanning too quickly. The **--min-rate** and **--max-rate** options are designed for these situations.

When the **--min-rate** option is given Nmap will do its best to send packets as fast as or faster than the given rate. The argument is a positive real number representing a packet rate in packets per second. For example, specifying **--min-rate 300** means that Nmap will try to keep the sending rate at or above 300 packets per second. Specifying a minimum rate does not keep Nmap from going faster if conditions warrant.

Likewise, **--max-rate** limits a scan's sending rate to a given maximum. Use **--max-rate 100**, for example, to limit sending to 100 packets per second on a fast network. Use **--max-rate 0.1** for a slow scan of one packet every ten seconds. Use **--min-rate** and **--max-rate** together to keep the rate inside a certain range.

These two options are global, affecting an entire scan, not individual hosts. They only affect port scans and host discovery scans. Other features like OS detection implement their own timing.

There are two conditions when the actual scanning rate may fall below the requested minimum. The first is if the minimum is faster than the fastest rate at which Nmap can send, which is dependent on hardware. In this case Nmap will simply send packets as fast as possible, but be aware that such high rates are likely to cause a loss of accuracy. The second case is when Nmap has nothing to send, for example at the end of a scan when the last probes have been sent and Nmap is waiting for them to time out or be responded to. It's normal to see the scanning rate drop at the end of a scan or in between hostgroups. The sending rate may temporarily exceed the maximum to make up for unpredictable delays, but on average the rate will stay at or below the maximum.

Specifying a minimum rate should be done with care. Scanning faster than a network can support may lead to a loss of accuracy. In some cases, using a faster rate can make a scan take *longer* than it would with a slower rate. This is because Nmap's

adaptive retransmission algorithms will detect the network congestion caused by an excessive scanning rate and increase the number of retransmissions in order to improve accuracy. So even though packets are sent at a higher rate, more packets are sent overall. Cap the number of retransmissions with the **--max-retries** option if you need to set an upper limit on total scan time.

--defeat-rst-ratelimit .

Many hosts have long used rate limiting. to reduce the number of ICMP error messages (such as

port-unreachable errors) they send. Some systems now apply similar rate limits to the RST (reset) packets they generate. This can slow Nmap down dramatically as it adjusts its timing to reflect those rate limits. You can tell Nmap to ignore those rate limits (for port scans such as SYN scan which *don't* treat non-responsive ports as open) by specifying **--defeat-rst-ratelimit**.

Using this option can reduce accuracy, as some ports will appear non-responsive because Nmap didn't wait long enough for a rate-limited RST response. With a SYN scan, the non-response results in the port being labeled filtered rather than the closed state we see when RST packets are received. This option is useful when you only care about open ports, and distinguishing between closed and filtered ports isn't worth the extra time.

--nsock-engine epoll|kqueue|poll|select .

Enforce use of a given nsock IO multiplexing engine. Only the select(2)-based fallback engine is guaranteed to be available on your system. Engines are named after the name of the IO management facility they leverage. Engines currently implemented are epoll, kqueue, poll, and select, but not all will be present on any platform. Use **nmap -V** to see which engines are supported.

-T paranoid|sneaky|polite|normal|aggressive|insane (Set a timing template) .

While the fine-grained timing controls discussed in the previous section are powerful and effective, some people find them confusing. Moreover, choosing the appropriate values can sometimes take more time than the scan you are trying to optimize. So Nmap offers a simpler approach, with six timing templates. You can specify them with the **-T** option and their number (0-5) or their name. The template names are **paranoid (0)**, **sneaky (1)**, **polite (2)**, **normal (3)**, **aggressive (4)**, and **insane (5)**. The first two are for IDS evasion. Polite mode slows down the scan to use less bandwidth and target machine resources. Normal mode is the default and so **-T3** does nothing. Aggressive mode speeds scans up by making the assumption that you are on a reasonably fast and reliable network. Finally insane mode. assumes that you are on an extraordinarily fast network or are willing to sacrifice some accuracy for speed.

These templates allow the user to specify how aggressive they wish to be, while leaving Nmap to pick the exact timing values. The templates also make some minor speed adjustments for which fine-grained control options do not currently exist. For example, **-T4** prohibits the dynamic scan delay from exceeding 10 ms for TCP ports and **-T5** caps that value at 5 ms. Templates can be used in combination with fine-grained controls, and the fine-grained controls will you specify will take precedence over the timing template default for that parameter. I recommend using **-T4** when scanning reasonably modern and reliable networks. Keep that option even when you add fine-grained controls so that you benefit from those extra minor optimizations that it enables.

If you are on a decent broadband or ethernet connection, I would recommend always using **-T4**. Some people love **-T5** though it is too aggressive for my taste. People sometimes specify **-T2** because they think it is less likely to crash hosts or because they consider themselves to be polite in general. They often don't realize just how slow **-T polite** really is. Their scan may take ten times longer than a default scan. Machine crashes and bandwidth problems are rare with the default timing options (**-T3**) and so I normally recommend that for cautious scanners. Omitting version detection is far more effective than playing with timing values at reducing these problems.

While **-T0** and **-T1** may be useful for avoiding IDS alerts, they will take an extraordinarily long time to scan thousands of machines or ports. For such a long scan, you may prefer to set the exact timing values you need rather than rely on the canned **-T0** and **-T1** values.

The main effects of **T0** are serializing the scan so only one port is scanned at a time, and waiting five minutes between sending each probe. **T1** and **T2** are similar but they only wait 15 seconds and 0.4 seconds, respectively, between probes. **T3** is Nmap's default behavior, which includes parallelization. **-T4** does the equivalent of **--max-rtt-timeout 1250ms --initial-rtt-timeout 500ms --max-retries 6** and sets the maximum TCP scan delay to 10 milliseconds. **T5** does the equivalent of **--max-rtt-timeout 300ms --min-rtt-timeout 50ms --initial-rtt-timeout 250ms**

--max-retries 2 --host-timeout 15m as well as setting the maximum TCP scan delay to 5 ms.

FIREWALL/IDS EVASION AND SPOOFING

Many Internet pioneers envisioned a global open network with a universal IP address space allowing virtual connections between any two nodes. This allows hosts to act as true peers, serving and retrieving information from each other. People could access all of their home systems from work, changing the climate control settings or unlocking the doors for early guests. This vision of universal connectivity has been stifled by address space shortages and security concerns. In the early 1990s, organizations began deploying firewalls for the express purpose of reducing connectivity. Huge networks were cordoned off from the unfiltered Internet by application proxies, network address translation, and packet filters. The unrestricted flow of information gave way to tight regulation of approved communication channels and the content that passes over them.

Network obstructions such as firewalls can make mapping a network exceedingly difficult. It will not get any easier, as stifling casual reconnaissance is often a key goal of implementing the devices. Nevertheless, Nmap offers many features to help understand these complex networks, and to verify that filters are working as intended. It even supports mechanisms for bypassing poorly implemented defenses. One of the best methods of understanding your network security posture is to try to defeat it. Place yourself in the mind-set of an attacker, and deploy techniques from this section against your networks. Launch an FTP bounce scan, idle scan, fragmentation attack, or try to tunnel through one of your own proxies.

In addition to restricting network activity, companies are increasingly monitoring traffic with intrusion detection systems (IDS). All of the major IDSs ship with rules designed to detect Nmap scans because scans are sometimes a precursor to attacks. Many of these products have recently morphed into intrusion *prevention* systems (IPS). that actively block traffic deemed malicious. Unfortunately for network administrators and IDS vendors, reliably detecting bad intentions by analyzing packet data is a tough problem. Attackers with patience, skill, and the help of certain Nmap options can usually pass by IDSs undetected. Meanwhile, administrators must cope with large numbers of false positive results where innocent activity is misdiagnosed and alerted on or blocked.

Occasionally people suggest that Nmap should not offer features for evading firewall rules or sneaking past IDSs. They argue that these features are just as likely to be misused by attackers as used by administrators to enhance security. The problem with this logic is that these methods would still be used by attackers, who would just find other tools or patch the functionality into Nmap. Meanwhile, administrators would find it that much harder to do their jobs. Deploying only modern, patched FTP servers is a far more powerful defense than trying to prevent the distribution of tools implementing the FTP bounce attack.

There is no magic bullet (or Nmap option) for detecting and subverting firewalls and IDS systems. It takes skill and experience. A tutorial is beyond the scope of this reference guide, which only lists the relevant options and describes what they do.

-f (fragment packets); **--mtu** (using the specified MTU) .

The **-f** option causes the requested scan (including ping scans) to use tiny fragmented IP packets. The idea is to split up the TCP header over several packets to make it harder for packet filters, intrusion detection systems, and other annoyances to detect what you are doing. Be careful with this! Some programs have trouble handling these tiny packets. The old-school sniffer named Sniffit segmentation faulted immediately upon receiving the first fragment. Specify this option once, and Nmap splits the packets into eight bytes or less after the IP header. So a 20-byte TCP header would be split into three packets. Two with eight bytes of the TCP header, and one with the final four. Of course each fragment also has an IP header. Specify **-f** again to use 16 bytes per fragment (reducing the number of fragments).. Or you can specify your own offset size with the **--mtu** option. Don't also specify **-f** if you use **--mtu**. The offset must be a multiple of eight. While fragmented packets won't get by packet filters and firewalls that queue all IP fragments, such as the `CONFIG_IP_ALWAYS_DEFRAG` option in the Linux kernel, some networks can't afford the performance hit this causes and thus leave it disabled. Others can't enable this because fragments may take different routes into their networks. Some source systems defragment outgoing packets in the kernel. Linux with the iptables. connection tracking module is one such example. Do a scan while a sniffer such as Wireshark. is running to ensure that sent packets are fragmented. If your host OS is causing problems, try the **--send-eth**. option to

bypass the IP layer and send raw ethernet frames.

Fragmentation is only supported for Nmap's raw packet features, which includes TCP and UDP port scans (except connect scan and FTP bounce scan) and OS detection. Features such as version detection and the Nmap Scripting Engine generally don't support fragmentation because they rely on your host's TCP stack to communicate with target services.

-D *decoy1[,decoy2][,ME][,...]* (Cloak a scan with decoys) .

Causes a decoy scan to be performed, which makes it appear to the remote host that the host(s) you specify as decoys are scanning the target network too. Thus their IDS might report 5–10 port scans from unique IP addresses, but they won't know which IP was scanning them and which were innocent decoys. While this can be defeated through router path tracing, response–dropping, and other active mechanisms, it is generally an effective technique for hiding your IP address.

Separate each decoy host with commas, and you can optionally use ME. as one of the decoys to represent the position for your real IP address. If you put ME in the sixth position or later, some common port scan detectors (such as Solar Designer's. excellent Scanlogd). are unlikely to show your IP address at all. If you don't use ME, Nmap will put you in a random position. You can also use RND. to generate a random, non–reserved IP address, or RND:*number* to generate *number* addresses.

Note that the hosts you use as decoys should be up or you might accidentally SYN flood your targets. Also it will be pretty easy to determine which host is scanning if only one is actually up on the network. You might want to use IP addresses instead of names (so the decoy networks don't see you in their nameserver logs).

Decoys are used both in the initial ping scan (using ICMP, SYN, ACK, or whatever) and during the actual port scanning phase. Decoys are also used during remote OS detection (**-O**). Decoys do not work with version detection or TCP connect scan. When a scan delay is in effect, the delay is enforced between each batch of spoofed probes, not between each individual probe. Because decoys are sent as a batch all at once, they may temporarily violate congestion control limits.

It is worth noting that using too many decoys may slow your scan and potentially even make it less accurate. Also, some ISPs will filter out your spoofed packets, but many do not restrict spoofed IP packets at all.

-S *IP_Address* (Spoof source address) .

In some circumstances, Nmap may not be able to determine your source address (Nmap will tell you if this is the case). In this situation, use **-S** with the IP address of the interface you wish to send packets through.

Another possible use of this flag is to spoof the scan to make the targets think that *someone else* is scanning them. Imagine a company being repeatedly port scanned by a competitor! The **-e** option and **-Pn** are generally required for this sort of usage. Note that you usually won't receive reply packets back (they will be addressed to the IP you are spoofing), so Nmap won't produce useful reports.

-e *interface* (Use specified interface) .

Tells Nmap what interface to send and receive packets on. Nmap should be able to detect this automatically, but it will tell you if it cannot.

--source-port *portnumber*; **-g** *portnumber* (Spoof source port number) .

One surprisingly common misconfiguration is to trust traffic based only on the source port number. It is easy to understand how this comes about. An administrator will set up a shiny new firewall, only to be flooded with complaints from ungrateful users whose applications stopped working. In particular, DNS may be broken because the UDP DNS replies from external servers can no longer enter the network. FTP is another common example. In active FTP transfers, the remote server tries to establish a connection back to the client to transfer the requested file.

Secure solutions to these problems exist, often in the form of application-level proxies or protocol-parsing firewall modules. Unfortunately there are also easier, insecure solutions. Noting that DNS replies come from port 53 and active FTP from port 20, many administrators have fallen into the trap of simply allowing incoming traffic from those ports. They often assume that no attacker would notice and exploit such firewall holes. In other cases, administrators consider this a short-term stop-gap measure until they can implement a more secure solution. Then they forget the security upgrade.

Overworked network administrators are not the only ones to fall into this trap. Numerous products have shipped with these insecure rules. Even Microsoft has been guilty. The IPsec filters that shipped with Windows 2000 and Windows XP contain an implicit rule that allows all TCP or UDP traffic from port 88 (Kerberos). In another well-known case, versions of the Zone Alarm personal firewall up to 2.1.25 allowed any incoming UDP packets with the source port 53 (DNS) or 67 (DHCP).

Nmap offers the **-g** and **--source-port** options (they are equivalent) to exploit these weaknesses. Simply provide a port number and Nmap will send packets from that port where possible. Most scanning operations that use raw sockets, including SYN and UDP scans, support the option completely. The option notably doesn't have an effect for any operations that use normal operating system sockets, including DNS requests, TCP **connect** scan, version detection, and script scanning. Setting the source port also doesn't work for OS detection, because Nmap must use different port numbers for certain OS detection tests to work properly.

--data-length *number* (Append random data to sent packets) .

Normally Nmap sends minimalist packets containing only a header. So its TCP packets are generally 40 bytes and ICMP echo requests are just 28. Some UDP ports and IP protocols get a custom payload by default. This option tells Nmap to append the given number of random bytes to most of the packets it sends, and not to use any protocol-specific payloads. (Use **--data-length 0** for no random or protocol-specific payloads.. OS detection (**-O**) packets are not affected. because accuracy there requires probe consistency, but most pinging and portscan packets support this. It slows things down a little, but can make a scan slightly less conspicuous.

--ip-options *S*[*R* [*route*]]*L* [*route*]]*T**U* ... ; **--ip-options** *hex string* (Send packets with specified ip options) .

The **IP protocol**^[13] offers several options which may be placed in packet headers. Unlike the ubiquitous TCP options, IP options are rarely seen due to practicality and security concerns. In fact, many Internet routers block the most dangerous options such as source routing. Yet options can still be useful in some cases for determining and manipulating the network route to target machines. For example, you may be able to use the record route option to determine a path to a target even when more traditional traceroute-style approaches fail. Or if your packets are being dropped by a certain firewall, you may be able to specify a different route with the strict or loose source routing options.

The most powerful way to specify IP options is to simply pass in values as the argument to **--ip-options**. Precede each hex number with \x then the two digits. You may repeat certain characters by following them with an asterisk and then the number of times you wish them to repeat. For example, \x01\x07\x04\x00*36\x01 is a hex string containing 36 NUL bytes.

Nmap also offers a shortcut mechanism for specifying options. Simply pass the letter R, T, or U to request record-route, record-timestamp, or both options together, respectively. Loose or strict source routing may be specified with an L or S followed by a space and then a space-separated list of IP addresses.

If you wish to see the options in packets sent and received, specify **--packet-trace**. For more information and examples of using IP options with Nmap, see <http://seclists.org/nmap-dev/2006/q3/52>.

--ttl *value* (Set IP time-to-live field) .

Sets the IPv4 time-to-live field in sent packets to the given value.

--randomize-hosts (Randomize target host order) .

Tells Nmap to shuffle each group of up to 16384 hosts before it scans them. This can make the scans less obvious to various network monitoring systems, especially when you combine it with slow timing options. If you want to randomize over larger group sizes, increase *PING_GROUP_SZ.* in *nmap.h.* and recompile. An alternative solution is to generate the target IP list with a list scan (**-sL -n -oN filename**), randomize it with a Perl script, then provide the whole list to Nmap with **-iL.**

--spoof-mac *MAC address, prefix, or vendor name* (Spoof MAC address) .

Asks Nmap to use the given MAC address for all of the raw ethernet frames it sends. This option implies **--send-eth.** to ensure that Nmap actually sends ethernet-level packets. The MAC given can take several formats. If it is simply the number 0, Nmap chooses a completely random MAC address for the session. If the given string is an even number of hex digits (with the pairs optionally separated by a colon), Nmap will use those as the MAC. If fewer than 12 hex digits are provided, Nmap fills in the remainder of the six bytes with random values. If the argument isn't a zero or hex string, Nmap looks through *nmap-mac-prefixes* to find a vendor name containing the given string (it is case insensitive). If a match is found, Nmap uses the vendor's OUI (three-byte prefix). and fills out the remaining three bytes randomly. Valid **--spoof-mac** argument examples are Apple, 0, 01:02:03:04:05:06, deadbeefcfe, 0020F2, and Cisco. This option only affects raw packet scans such as SYN scan or OS detection, not connection-oriented features such as version detection or the Nmap Scripting Engine.

--badsum (Send packets with bogus TCP/UDP checksums) .

Asks Nmap to use an invalid TCP, UDP or SCTP checksum for packets sent to target hosts. Since virtually all host IP stacks properly drop these packets, any responses received are likely coming from a firewall or IDS that didn't bother to verify the checksum. For more details on this technique, see <http://nmap.org/p60-12.html>

--adler32 (Use deprecated Adler32 instead of CRC32C for SCTP checksums) .

Asks Nmap to use the deprecated Adler32 algorithm for calculating the SCTP checksum. If **--adler32** is not given, CRC-32C (Castagnoli) is used. [RFC 2960](#)^[14] originally defined Adler32 as checksum algorithm for SCTP; [RFC 4960](#)^[7] later redefined the SCTP checksums to use CRC-32C. Current SCTP implementations should be using CRC-32C, but in order to elicit responses from old, legacy SCTP implementations, it may be preferable to use Adler32.

OUTPUT

Any security tool is only as useful as the output it generates. Complex tests and algorithms are of little value if they aren't presented in an organized and comprehensible fashion. Given the number of ways Nmap is used by people and other software, no single format can please everyone. So Nmap offers several formats, including the interactive mode for humans to read directly and XML for easy parsing by software.

In addition to offering different output formats, Nmap provides options for controlling the verbosity of output as well as debugging messages. Output types may be sent to standard output or to named files, which Nmap can append to or clobber. Output files may also be used to resume aborted scans.

Nmap makes output available in five different formats. The default is called interactive output, and it is sent to standard output (stdout).. There is also normal output, which is similar to interactive except that it displays less runtime information and warnings since it is expected to be analyzed after the scan completes rather than interactively.

XML output. is one of the most important output types, as it can be converted to HTML, easily parsed by programs such as Nmap graphical user interfaces, or imported into databases.

The two remaining output types are the simple grepable output. which includes most information for a target host on a single line, and sCRiPt KiDDi3 OutPUt. for users who consider themselves |<-r4d.

While interactive output is the default and has no associated command-line options, the other four format options use the same syntax. They take one argument, which is the filename that results should be stored in. Multiple formats may be specified, but each format may only be specified once. For example, you may

wish to save normal output for your own review while saving XML of the same scan for programmatic analysis. You might do this with the options **-oX myscan.xml -oN myscan.nmap**. While this chapter uses the simple names like myscan.xml for brevity, more descriptive names are generally recommended. The names chosen are a matter of personal preference, though I use long ones that incorporate the scan date and a word or two describing the scan, placed in a directory named after the company I'm scanning.

While these options save results to files, Nmap still prints interactive output to stdout as usual. For example, the command **nmap -oX myscan.xml target** prints XML to myscan.xml and fills standard output with the same interactive results it would have printed if **-oX** wasn't specified at all. You can change this by passing a hyphen character as the argument to one of the format types. This causes Nmap to deactivate interactive output, and instead print results in the format you specified to the standard output stream. So the command **nmap -oX - target** will send only XML output to stdout.. Serious errors may still be printed to the normal error stream, stderr..

Unlike some Nmap arguments, the space between the logfile option flag (such as **-oX**) and the filename or hyphen is mandatory. If you omit the flags and give arguments such as **-oG-** or **-oXscan.xml**, a backwards compatibility feature of Nmap will cause the creation of *normal format* output files named G- and Xscan.xml respectively.

All of these arguments support **strftime**-like conversions in the filename. %H, %M, %S, %m, %d, %y, and %Y are all exactly the same as in **strftime**. %T is the same as %H%M%S, %R is the same as %H%M, and %D is the same as %m%d%y. A % followed by any other character just yields that character (%% gives you a percent symbol). So **-oX 'scan-%T-%D.xml'** will use an XML file with a name in the form of scan-144840-121307.xml.

Nmap also offers options to control scan verbosity and to append to output files rather than clobbering them. All of these options are described below.

Nmap Output Formats

-oN *filespec* (normal output) .

Requests that normal output be directed to the given filename. As discussed above, this differs slightly from interactive output.

-oX *filespec* (XML output) .

Requests that XML output be directed to the given filename. Nmap includes a document type definition (DTD) which allows XML parsers to validate Nmap XML output. While it is primarily intended for programmatic use, it can also help humans interpret Nmap XML output. The DTD defines the legal elements of the format, and often enumerates the attributes and values they can take on. The latest version is always available from <https://svn.nmap.org/nmap/docs/nmap.dtd>.

XML offers a stable format that is easily parsed by software. Free XML parsers are available for all major computer languages, including C/C++, Perl, Python, and Java. People have even written bindings for most of these languages to handle Nmap output and execution specifically. Examples are [Nmap::Scanner](#)^[15] and [Nmap::Parser](#)^[16] in Perl CPAN. In almost all cases that a non-trivial application interfaces with Nmap, XML is the preferred format.

The XML output references an XSL stylesheet which can be used to format the results as HTML. The easiest way to use this is simply to load the XML output in a web browser such as Firefox or IE. By default, this will only work on the machine you ran Nmap on (or a similarly configured one) due to the hard-coded nmap.xsl filesystem path. Use the **---webxml** or **---stylesheet** options to create portable XML files that render as HTML on any web-connected machine.

-oS *filespec* (ScRipT KIdd|3 oUTpuT) .

Script kiddie output is like interactive output, except that it is post-processed to better suit the 133t HaXXorZ who previously looked down on Nmap due to its consistent capitalization and spelling. Humor impaired people should note that this option is making fun of the script kiddies before flaming me for supposedly "helping them".

-oG *filespec* (greppable output) .

This output format is covered last because it is deprecated. The XML output format is far more powerful, and is nearly as convenient for experienced users. XML is a standard for which dozens of excellent parsers are available, while grepable output is my own simple hack. XML is extensible to support new Nmap features as they are released, while I often must omit those features from grepable output for lack of a place to put them.

Nevertheless, grepable output is still quite popular. It is a simple format that lists each host on one line and can be trivially searched and parsed with standard Unix tools such as `grep`, `awk`, `cut`, `sed`, `diff`, and `Perl`. Even I usually use it for one-off tests done at the command line. Finding all the hosts with the SSH port open or that are running Solaris takes only a simple `grep` to identify the hosts, piped to an `awk` or `cut` command to print the desired fields.

Grepable output consists of comments (lines starting with a pound (#)). and target lines. A target line includes a combination of six labeled fields, separated by tabs and followed with a colon. The fields are Host, Ports, Protocols, Ignored State, OS, Seq Index, IP ID, and Status.

The most important of these fields is generally Ports, which gives details on each interesting port. It is a comma separated list of port entries. Each port entry represents one interesting port, and takes the form of seven slash (/) separated subfields. Those subfields are: Port number, State, Protocol, Owner, Service, SunRPC info, and Version info.

As with XML output, this man page does not allow for documenting the entire format. A more detailed look at the Nmap grepable output format is available from <http://nmap.org/book/output-formats-grepable-output.html>.

-oA *basename* (Output to all formats) .

As a convenience, you may specify **-oA** *basename* to store scan results in normal, XML, and grepable formats at once. They are stored in *basename.nmap*, *basename.xml*, and *basename.gnmap*, respectively. As with most programs, you can prefix the filenames with a directory path, such as `~/nmaplogs/foocorp/` on Unix or `c:\hacking\sco` on Windows.

Verbosity and debugging options

-v (Increase verbosity level) .

Increases the verbosity level, causing Nmap to print more information about the scan in progress. Open ports are shown as they are found and completion time estimates are provided when Nmap thinks a scan will take more than a few minutes. Use it twice or more for even greater verbosity: **-vv**, or give a verbosity level directly, for example **-v3**.

Most changes only affect interactive output, and some also affect normal and script kiddie output. The other output types are meant to be processed by machines, so Nmap can give substantial detail by default in those formats without fatiguing a human user. However, there are a few changes in other modes where output size can be reduced substantially by omitting some detail. For example, a comment line in the grepable output that provides a list of all ports scanned is only printed in verbose mode because it can be quite long.

-d (Increase debugging level) .

When even verbose mode doesn't provide sufficient data for you, debugging is available to flood you with much more! As with the verbosity option (**-v**), debugging is enabled with a command-line flag (**-d**) and the debug level can be increased by specifying it multiple times, as in **-dd**, or by setting a level directly. For example, **-d9** sets level nine. That is the highest effective level and will produce thousands of lines unless you run a very simple scan with very few ports and targets.

Debugging output is useful when a bug is suspected in Nmap, or if you are simply confused as to what Nmap is doing and why. As this feature is mostly intended for developers, debug lines aren't always self-explanatory. You may get something like: Timeout vals: srtr: -1 rttvar: -1 to: 1000000 delta 14987 ==> srtr: 14987 rttvar: 14987 to: 100000. If you don't understand a line, your only recourse is

to ignore it, look it up in the source code, or request help from the development list (nmap-dev).. Some lines are self explanatory, but the messages become more obscure as the debug level is increased.

--reason (Host and port state reasons) .

Shows the reason each port is set to a specific state and the reason each host is up or down. This option displays the type of the packet that determined a port or hosts state. For example, A RST packet from a closed port or an echo reply from an alive host. The information Nmap can provide is determined by the type of scan or ping. The SYN scan and SYN ping (**-sS** and **-PS**) are very detailed, but the TCP connect scan (**-sT**) is limited by the implementation of the **connect** system call. This feature is automatically enabled by the debug option (**-d**). and the results are stored in XML log files even if this option is not specified.

--stats-every time (Print periodic timing stats) .

Periodically prints a timing status message after each interval of *time*. The time is a specification of the kind described in the section called "TIMING AND PERFORMANCE"; so for example, use **--stats-every 10s** to get a status update every 10 seconds. Updates are printed to interactive output (the screen) and XML output.

--packet-trace (Trace packets and data sent and received) .

Causes Nmap to print a summary of every packet sent or received. This is often used for debugging, but is also a valuable way for new users to understand exactly what Nmap is doing under the covers. To avoid printing thousands of lines, you may want to specify a limited number of ports to scan, such as **-p20-30**. If you only care about the goings on of the version detection subsystem, use **--version-trace** instead. If you only care about script tracing, specify **--script-trace**. With **--packet-trace**, you get all of the above.

--open (Show only open (or possibly open) ports) .

Sometimes you only care about ports you can actually connect to (open ones), and don't want results cluttered with closed, filtered, and closed|filtered ports. Output customization is normally done after the scan using tools such as grep, awk, and Perl, but this feature was added due to overwhelming requests. Specify **--open** to only see hosts with at least one open, open|filtered, or unfiltered port, and only see ports in those states. These three states are treated just as they normally are, which means that open|filtered and unfiltered may be condensed into counts if there are an overwhelming number of them.

--iflist (List interfaces and routes) .

Prints the interface list and system routes as detected by Nmap. This is useful for debugging routing problems or device mischaracterization (such as Nmap treating a PPP connection as ethernet).

Miscellaneous output options

--append-output (Append to rather than clobber output files) .

When you specify a filename to an output format flag such as **-oX** or **-oN**, that file is overwritten by default. If you prefer to keep the existing content of the file and append the new results, specify the **--append-output** option. All output filenames specified in that Nmap execution will then be appended to rather than clobbered. This doesn't work well for XML (**-oX**) scan data as the resultant file generally won't parse properly until you fix it up by hand.

--resume filename (Resume aborted scan) .

Some extensive Nmap runs take a very long time—on the order of days. Such scans don't always run to completion. Restrictions may prevent Nmap from being run during working hours, the network could go down, the machine Nmap is running on might suffer a planned or unplanned reboot, or Nmap itself could crash. The administrator running Nmap could cancel it for any other reason as well, by pressing ctrl-C. Restarting the whole scan from the beginning may be undesirable. Fortunately, if normal (**-oN**) or grepable (**-oG**) logs were kept, the user can ask Nmap to resume scanning with the target it was working on when execution ceased. Simply specify the **--resume** option and pass the normal/grepable output file as its argument. No other arguments are permitted, as Nmap parses the output file to use the same ones specified previously. Simply call Nmap as **nmap --resume**

logfile. Nmap will append new results to the data files specified in the previous execution. Resumption does not support the XML output format because combining the two runs into one valid XML file would be difficult.

--stylesheet *path or URL* (Set XSL stylesheet to transform XML output) .

Nmap ships with an XSL stylesheet named `nmap.xsl` for viewing or translating XML output to HTML. The XML output includes an `xml-stylesheet` directive which points to `nmap.xml` where it was initially installed by Nmap. Run the XML file through an XSLT processor such as [xsltproc](#)^[17] to produce an HTML file. Directly opening the XML file in a browser no longer works well because modern browsers limit the locations a stylesheet may be loaded from. If you wish to use a different stylesheet, specify it as the argument to **--stylesheet**. You must pass the full pathname or URL. One common invocation is **--stylesheet http://nmap.org/svn/docs/nmap.xsl**. This tells an XSLT processor to load the latest version of the stylesheet from Nmap.Org. The **--webxml** option does the same thing with less typing and memorization. Loading the XSL from Nmap.Org makes it easier to view results on a machine that doesn't have Nmap (and thus `nmap.xsl`) installed. So the URL is often more useful, but the local filesystem location of `nmap.xsl` is used by default for privacy reasons.

--webxml (Load stylesheet from Nmap.Org) .

This is a convenience option, nothing more than an alias for **--stylesheet http://nmap.org/svn/docs/nmap.xsl**.

--no-stylesheet (Omit XSL stylesheet declaration from XML) .

Specify this option to prevent Nmap from associating any XSL stylesheet with its XML output. The `xml-stylesheet` directive is omitted.

MISCELLANEOUS OPTIONS

This section describes some important (and not-so-important) options that don't really fit anywhere else.

-6 (Enable IPv6 scanning) .

Nmap has IPv6 support for its most popular features. Ping scanning, port scanning, version detection, and the Nmap Scripting Engine all support IPv6. The command syntax is the same as usual except that you also add the **-6** option. Of course, you must use IPv6 syntax if you specify an address rather than a hostname. An address might look like `3ffe:7501:4819:2000:210:f3ff:fe03:14d0`, so hostnames are recommended. The output looks the same as usual, with the IPv6 address on the "interesting ports" line being the only IPv6 giveaway.

While IPv6 hasn't exactly taken the world by storm, it gets significant use in some (usually Asian) countries and most modern operating systems support it. To use Nmap with IPv6, both the source and target of your scan must be configured for IPv6. If your ISP (like most of them) does not allocate IPv6 addresses to you, free tunnel brokers are widely available and work fine with Nmap. I use the free IPv6 tunnel broker service at <http://www.tunnelbroker.net>. Other tunnel brokers are [listed at Wikipedia](#)^[18]. 6to4 tunnels are another popular, free approach.

On Windows, raw-socket IPv6 scans are supported only on ethernet devices (not tunnels), and only on Windows Vista and later. Use the **--unprivileged** option in other situations.

-A (Aggressive scan options) .

This option enables additional advanced and aggressive options. I haven't decided exactly which it stands for yet. Presently this enables OS detection (**-O**), version scanning (**-sV**), script scanning (**-sC**) and traceroute (**--traceroute**). More features may be added in the future. The point is to enable a comprehensive set of scan options without people having to remember a large set of flags. However, because script scanning with the default set is considered intrusive, you should not use **-A** against target networks without permission. This option only enables features, and not timing options (such as **-T4**) or verbosity options (**-v**) that you might want as well.

--datadir *directoryname* (Specify custom Nmap data file location) .

Nmap obtains some special data at runtime in files named `nmap-service-probes`, `nmap-services`, `nmap-protocols`, `nmap-rpc`, `nmap-mac-prefixes`, and `nmap-os-db`. If the location of any of these files has been specified (using the **--servicedb** or **--versiondb** options), that location is used for that

file. After that, Nmap searches these files in the directory specified with the **---datadir** option (if any). Any files not found there, are searched for in the directory specified by the **NMAPDIR** environment variable. Next comes `~/nmap` for real and effective UIDs; or on Windows, `HOME\AppData\Roaming\nmap` (where *HOME* is the user's home directory, like `C:\Users\user`). This is followed by the location of the nmap executable and the same location with `../share/nmap` appended. Then a compiled-in location such as `/usr/local/share/nmap` or `/usr/share/nmap`.

---servicedb *services file* (Specify custom services file) .

Asks Nmap to use the specified services file rather than the `nmap-services` data file that comes with Nmap. Using this option also causes a fast scan (**-F**) to be used. See the description for **---datadir** for more information on Nmap's data files.

---versiondb *service probes file* (Specify custom service probes file) .

Asks Nmap to use the specified service probes file rather than the `nmap-service-probes` data file that comes with Nmap. See the description for **---datadir** for more information on Nmap's data files.

---send-eth (Use raw ethernet sending) .

Asks Nmap to send packets at the raw ethernet (data link) layer rather than the higher IP (network) layer. By default, Nmap chooses the one which is generally best for the platform it is running on. Raw sockets (IP layer) are generally most efficient for Unix machines, while ethernet frames are required for Windows operation since Microsoft disabled raw socket support. Nmap still uses raw IP packets on Unix despite this option when there is no other choice (such as non-ethernet connections).

---send-ip (Send at raw IP level) .

Asks Nmap to send packets via raw IP sockets rather than sending lower level ethernet frames. It is the complement to the **---send-eth** option discussed previously.

---privileged (Assume that the user is fully privileged) .

Tells Nmap to simply assume that it is privileged enough to perform raw socket sends, packet sniffing, and similar operations that usually require root privileges. on Unix systems. By default Nmap quits if such operations are requested but `geteuid` is not zero. **---privileged** is useful with Linux kernel capabilities and similar systems that may be configured to allow unprivileged users to perform raw-packet scans. Be sure to provide this option flag before any flags for options that require privileges (SYN scan, OS detection, etc.). The **NMAP_PRIVILEGED** environment variable may be set as an equivalent alternative to **---privileged**.

---unprivileged (Assume that the user lacks raw socket privileges) .

This option is the opposite of **---privileged**. It tells Nmap to treat the user as lacking network raw socket and sniffing privileges. This is useful for testing, debugging, or when the raw network functionality of your operating system is somehow broken. The **NMAP_UNPRIVILEGED** environment variable may be set as an equivalent alternative to **---unprivileged**.

---release-memory (Release memory before quitting) .

This option is only useful for memory-leak debugging. It causes Nmap to release allocated memory just before it quits so that actual memory leaks are easier to spot. Normally Nmap skips this as the OS does this anyway upon process termination.

-V; **---version** (Print version number) .

Prints the Nmap version number and exits.

-h; **---help** (Print help summary page) .

Prints a short help screen with the most common command flags. Running Nmap without any arguments does the same thing.

RUNTIME INTERACTION

During the execution of Nmap, all key presses are captured. This allows you to interact with the program without aborting and restarting it. Certain special keys will change options, while any other keys will print out a status message telling you about the scan. The convention is that *lowercase letters increase* the amount of printing, and *uppercase letters decrease* the printing. You may also press `?` for help.

v / V

Increase / decrease the verbosity level

d / D

Increase / decrease the debugging Level

p / P

Turn on / off packet tracing

?

Print a runtime interaction help screen

Anything else

Print out a status message like this:

Stats: 0:00:07 elapsed; 20 hosts completed (1 up), 1 undergoing Service Scan

Service scan Timing: About 33.33% done; ETC: 20:57 (0:00:12 remaining)

EXAMPLES

Here are some Nmap usage examples, from the simple and routine to a little more complex and esoteric. Some actual IP addresses and domain names are used to make things more concrete. In their place you should substitute addresses/names from *your own network*. While I don't think port scanning other networks is or should be illegal, some network administrators don't appreciate unsolicited scanning of their networks and may complain. Getting permission first is the best approach.

For testing purposes, you have permission to scan the host scanme.nmap.org.. This permission only includes scanning via Nmap and not testing exploits or denial of service attacks. To conserve bandwidth, please do not initiate more than a dozen scans against that host per day. If this free scanning target service is abused, it will be taken down and Nmap will report Failed to resolve given hostname/IP: scanme.nmap.org. These permissions also apply to the hosts scanme2.nmap.org, scanme3.nmap.org, and so on, though those hosts do not currently exist.

nmap -v scanme.nmap.org

This option scans all reserved TCP ports on the machine scanme.nmap.org . The **-v** option enables verbose mode.

nmap -sS -O scanme.nmap.org/24

Launches a stealth SYN scan against each machine that is up out of the 256 IPs on the class C sized network where Scanme resides. It also tries to determine what operating system is running on each host that is up and running. This requires root privileges because of the SYN scan and OS detection.

nmap -sV -p 22,53,110,143,4564 198.116.0-255.1-127

Launches host enumeration and a TCP scan at the first half of each of the 255 possible eight-bit subnets in the 198.116 class B address space. This tests whether the systems run SSH, DNS, POP3, or IMAP on their standard ports, or anything on port 4564. For any of these ports found open, version detection is used to determine what application is running.

nmap -v -iR 100000 -Pn -p 80

Asks Nmap to choose 100,000 hosts at random and scan them for web servers (port 80). Host enumeration is disabled with **-Pn** since first sending a couple probes to determine whether a host is up is wasteful when you are only probing one port on each target host anyway.

nmap -Pn -p80 -oX logs/pb-port80scan.xml -oG logs/pb-port80scan.gnmap 216.163.128.20/20

This scans 4096 IPs for any web servers (without pinging them) and saves the output in grepable and XML formats.

NMAP BOOK

While this reference guide details all material Nmap options, it can't fully demonstrate how to apply those features to quickly solve real-world tasks. For that, we released Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning. Topics include subverting firewalls and intrusion detection systems, optimizing Nmap performance, and automating common networking tasks

with the Nmap Scripting Engine. Hints and instructions are provided for common Nmap tasks such as taking network inventory, penetration testing, detecting rogue wireless access points, and quashing network worm outbreaks. Examples and diagrams show actual communication on the wire. More than half of the book is available free online. See <http://nmap.org/book> for more information.

BUGS

Like its author, Nmap isn't perfect. But you can help make it better by sending bug reports or even writing patches. If Nmap doesn't behave the way you expect, first upgrade to the latest version available from <http://nmap.org>. If the problem persists, do some research to determine whether it has already been discovered and addressed. Try searching for the error message on our search page at <http://insecure.org/search.html> or at Google. Also try browsing the nmap-dev archives at <http://seclists.org/>. Read this full manual page as well. If nothing comes of this, mail a bug report to nmap-dev@insecure.org. Please include everything you have learned about the problem, as well as what version of Nmap you are running and what operating system version it is running on. Problem reports and Nmap usage questions sent to nmap-dev@insecure.org are far more likely to be answered than those sent to Fyodor directly. If you subscribe to the nmap-dev list before posting, your message will bypass moderation and get through more quickly. Subscribe at <http://cgi.insecure.org/mailman/listinfo/nmap-dev>.

Code patches to fix bugs are even better than bug reports. Basic instructions for creating patch files with your changes are available at <https://svn.nmap.org/nmap/HACKING>. Patches may be sent to nmap-dev@insecure.org (recommended) or to Fyodor directly.

AUTHOR

Gordon "Fyodor" Lyon fyodor@insecure.org (<http://insecure.org>)

Hundreds of people have made valuable contributions to Nmap over the years. These are detailed in the CHANGELOG. file which is distributed with Nmap and also available from <http://nmap.org/changelog.html>.

LEGAL NOTICES

Nmap Copyright and Licensing

The Nmap Security Scanner is (C) 1996–2012 Insecure.Com LLC. Nmap is also a registered trademark of Insecure.Com LLC. This program is free software; you may redistribute and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; Version 2 with the clarifications and exceptions described below. This guarantees your right to use, modify, and redistribute this software under certain conditions. If you wish to embed Nmap technology into proprietary software, we sell alternative licenses (contact sales@insecure.com). Dozens of software vendors already license Nmap technology such as host discovery, port scanning, OS detection, and version detection.

Note that the GPL places important restrictions on "derived works", yet it does not provide a detailed definition of that term. To avoid misunderstandings, we consider an application to constitute a "derivative work" for the purpose of this license if it does any of the following:

- Integrates source code from Nmap
- Reads or includes Nmap copyrighted data files, such as `nmap-os-db` or `nmap-service-probes`.
- Executes Nmap and parses the results (as opposed to typical shell or execution-menu apps, which simply display raw Nmap output and so are not derivative works.)
- Integrates/includes/aggregates Nmap into a proprietary executable installer, such as those produced by InstallShield.
- Links to a library or executes a program that does any of the above.

The term "Nmap" should be taken to also include any portions or derived works of Nmap. This list is not exclusive, but is meant to clarify our interpretation of derived works with some common examples. Our interpretation applies only to Nmap—we don't speak for other people's GPL works.

If you have any questions about the GPL licensing restrictions on using Nmap in non-GPL works, we would be happy to help. As mentioned above, we also offer alternative license to integrate Nmap into

proprietary applications and appliances. These contracts have been sold to many security vendors, and generally include a perpetual license as well as providing for priority support and updates as well as helping to fund the continued development of Nmap technology. Please email sales@insecure.com for further information.

As a special exception to the GPL terms, Insecure.Com LLC grants permission to link the code of this program with any version of the OpenSSL library which is distributed under a license identical to that listed in the included COPYING.OpenSSL file, and distribute linked combinations including the two.. You must obey the GNU GPL in all respects for all of the code used other than OpenSSL. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so.

If you received these files with a written license agreement or contract stating terms other than the terms above, then that alternative license agreement takes precedence over these comments.

Creative Commons License for this Nmap Guide

This Nmap Reference Guide is (C) 2005–2012 Insecure.Com LLC. It is hereby placed under version 3.0 of the [Creative Commons Attribution License](#)^[19]. This allows you redistribute and modify the work as you desire, as long as you credit the original source. Alternatively, you may choose to treat this document as falling under the same license as Nmap itself (discussed previously).

Source Code Availability and Community Contributions

Source is provided to this software because we believe users have a right to know exactly what a program is going to do before they run it. This also allows you to audit the software for security holes (none have been found so far).

Source code also allows you to port Nmap to new platforms, fix bugs, and add new features. You are highly encouraged to send your changes to nmap-dev@insecure.org for possible incorporation into the main distribution. By sending these changes to Fyodor or one of the Insecure.Org development mailing lists, it is assumed that you are offering the Nmap Project (Insecure.Com LLC) the unlimited, non-exclusive right to reuse, modify, and relicense the code. Nmap will always be available open source,. but this is important because the inability to relicense code has caused devastating problems for other Free Software projects (such as KDE and NASM). We also occasionally relicense the code to third parties as discussed above. If you wish to specify special license conditions of your contributions, just say so when you send them.

No Warranty.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License v2.0 for more details at <http://www.gnu.org/licenses/gpl-2.0.html>, or in the COPYING file included with Nmap.

It should also be noted that Nmap has occasionally been known to crash poorly written applications, TCP/IP stacks, and even operating systems.. While this is extremely rare, it is important to keep in mind. *Nmap should never be run against mission critical systems* unless you are prepared to suffer downtime. We acknowledge here that Nmap may crash your systems or networks and we disclaim all liability for any damage or problems Nmap could cause.

Inappropriate Usage

Because of the slight risk of crashes and because a few black hats like to use Nmap for reconnaissance prior to attacking systems, there are administrators who become upset and may complain when their system is scanned. Thus, it is often advisable to request permission before doing even a light scan of a network.

Nmap should never be installed with special privileges (e.g. `sudo` root).. That would open up a major security vulnerability as other users on the system (or attackers) could use it for privilege escalation.

Third-Party Software and Funding Notices

This product includes software developed by the [Apache Software Foundation](#)^[20]. A modified version of the [Libpcap portable packet capture library](#)^[21], is distributed along with Nmap. The Windows version of Nmap utilized the Libpcap-derived [WinPcap library](#)^[22], instead. Regular expression support is provided by the [PCRE library](#)^[23], which is open-source software, written by Philip Hazel.. Certain raw networking functions use the [Libdnet](#)^[24], networking library, which was written by Dug Song.. A

modified version is distributed with Nmap. Nmap can optionally link with the [OpenSSL cryptography toolkit](#)^[25] for SSL version detection support. The Nmap Scripting Engine uses an embedded version of the [Lua programming language](#)^[26]. The [Liblinear linear classification library](#)^[27] is used for our [IPv6 OS detection machine learning techniques](#)^[28].

All of the third-party software described in this paragraph is freely redistributable under BSD-style software licenses.

Binary packages for Windows and Mac OS X include support libraries necessary to run Zenmap and Ndiff with Python and PyGTK. (Unix platforms commonly make these libraries easy to install, so they are not part of the packages.) A listing of these support libraries and their licenses is included in the LICENSES files.

This software was supported in part through the [Google Summer of Code](#)^[29] and the [DARPA CINDER program](#)^[30] (DARPA-BAA-10-84).

United States Export Control.

Nmap only uses encryption when compiled with the optional OpenSSL support and linked with OpenSSL. When compiled without OpenSSL support, Insecure.Com LLC believes that Nmap is not subject to U.S. [Export Administration Regulations \(EAR\)](#)^[31] export control. As such, there is no applicable ECCN (export control classification number) and exportation does not require any special license, permit, or other governmental authorization.

When compiled with OpenSSL support or distributed as source code, Insecure.Com LLC believes that Nmap falls under U.S. ECCN [5D002](#)^[32] ("Information Security Software"). We distribute Nmap under the TSU exception for publicly available encryption software defined in [EAR 740.13\(e\)](#)^[33].

NOTES

1. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning
<http://nmap.org/book/>
2. RFC 1122
<http://www.rfc-editor.org/rfc/rfc1122.txt>
3. RFC 792
<http://www.rfc-editor.org/rfc/rfc792.txt>
4. RFC 950
<http://www.rfc-editor.org/rfc/rfc950.txt>
5. RFC 1918
<http://www.rfc-editor.org/rfc/rfc1918.txt>
6. UDP
<http://www.rfc-editor.org/rfc/rfc768.txt>
7. SCTP
<http://www.rfc-editor.org/rfc/rfc4960.txt>
8. TCP RFC
<http://www.rfc-editor.org/rfc/rfc793.txt>
9. RFC 959
<http://www.rfc-editor.org/rfc/rfc959.txt>
10. RFC 1323
<http://www.rfc-editor.org/rfc/rfc1323.txt>
11. Lua programming language
<http://lua.org>
12. precedence
<http://www.lua.org/manual/5.1/manual.html#2.5.3>

13. IP protocol
<http://www.rfc-editor.org/rfc/rfc791.txt>
14. RFC 2960
<http://www.rfc-editor.org/rfc/rfc2960.txt>
15. Nmap::Scanner
<http://sourceforge.net/projects/nmap-scanner/>
16. Nmap::Parser
<http://nmapparser.wordpress.com/>
17. xsltproc
<http://xmlsoft.org/XSLT/>
18. listed at Wikipedia
http://en.wikipedia.org/wiki/List_of_IPv6_tunnel_brokers
19. Creative Commons Attribution License
<http://creativecommons.org/licenses/by/3.0/>
20. Apache Software Foundation
<http://www.apache.org>
21. Libpcap portable packet capture library
<http://www.tcpdump.org>
22. WinPcap library
<http://www.winpcap.org>
23. PCRE library
<http://www.pcre.org>
24. Libdnet
<http://libdnet.sourceforge.net>
25. OpenSSL cryptography toolkit
<http://www.openssl.org>
26. Lua programming language
<http://www.lua.org>
27. Liblinear linear classification library
<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>
28. IPv6 OS detection machine learning techniques
<http://nmap.org/book/osdetect-guess.html#osdetect-guess-ipv6>
29. Google Summer of Code
<http://nmap.org/soc/>
30. DARPA CINDER program
https://www.fbo.gov/index?s=opportunity&mode=form&id=585e02a51f77af5cb3c9e06b9cc82c48&tab=core&_cvie
31. Export Administration Regulations (EAR)
http://www.access.gpo.gov/bis/ear/ear_data.html
32. 5D002
<http://www.access.gpo.gov/bis/ear/pdf/ccl5-pt2.pdf>
33. EAR 740.13(e)
<http://www.access.gpo.gov/bis/ear/pdf/740.pdf>

NAME

nologin – issue a security violation and exit

SYNOPSIS

edrc/bin/nologin

AVAILABILITY

WA2L/edrc

DESCRIPTION

issue a security violation message and exit.

The intention of this command is to forbid system login for certain user accounts by using this command instead of a login shell in **/etc/passwd(4)** or the related name service.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

4 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **passwd(4)**

NOTES

-

BUGS

-

AUTHOR

nologin was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

nping – Network packet generation tool / ping utility

SYNOPSIS

nping [*Options*] {*targets*}

DESCRIPTION

Nping is an open-source tool for network packet generation, response analysis and response time measurement. Nping allows users to generate network packets of a wide range of protocols, letting them tune virtually any field of the protocol headers. While Nping can be used as a simple ping utility to detect active hosts, it can also be used as a raw packet generator for network stack stress tests, ARP poisoning, Denial of Service attacks, route tracing, and other purposes.

Additionally, Nping offers a special mode of operation called the "Echo Mode", that lets users see how the generated probes change in transit, revealing the differences between the transmitted packets and the packets received at the other end. See section "Echo Mode" for details.

The output from Nping is a list of the packets that are being sent and received. The level of detail depends on the options used.

A typical Nping execution is shown in Example 1. The only Nping arguments used in this example are **-c**, to specify the number of times to target each host, **---tcp** to specify TCP Probe Mode, **-p 80,433** to specify the target ports; and then the two target hostnames.

Example 1. A representative Nping execution

```
# nping -c 1 ---tcp -p 80,433 scanme.nmap.org google.com
```

Starting Nping (<http://nmap.org/nping>)

```
SENT (0.0120s) TCP 96.16.226.135:50091 > 64.13.134.52:80 S ttl=64 id=52072 iplen=40 seq=1077657388 win=1480
RCVD (0.1810s) TCP 64.13.134.52:80 > 96.16.226.135:50091 SA ttl=53 id=0 iplen=44 seq=4158134847 win=5840 <m
SENT (1.0140s) TCP 96.16.226.135:50091 > 74.125.45.100:80 S ttl=64 id=13932 iplen=40 seq=1077657388 win=1480
RCVD (1.1370s) TCP 74.125.45.100:80 > 96.16.226.135:50091 SA ttl=52 id=52913 iplen=44 seq=2650443864 win=57:
SENT (2.0140s) TCP 96.16.226.135:50091 > 64.13.134.52:433 S ttl=64 id=8373 iplen=40 seq=1077657388 win=1480
SENT (3.0140s) TCP 96.16.226.135:50091 > 74.125.45.100:433 S ttl=64 id=23624 iplen=40 seq=1077657388 win=1480
```

Statistics for host scanme.nmap.org (64.13.134.52):

```
| Probes Sent: 2 | Rcvd: 1 | Lost: 1 (50.00%)
|_ Max rtt: 169.720ms | Min rtt: 169.720ms | Avg rtt: 169.720ms
```

Statistics for host google.com (74.125.45.100):

```
| Probes Sent: 2 | Rcvd: 1 | Lost: 1 (50.00%)
|_ Max rtt: 122.686ms | Min rtt: 122.686ms | Avg rtt: 122.686ms
```

Raw packets sent: 4 (160B) | Rcvd: 2 (92B) | Lost: 2 (50.00%)

Tx time: 3.00296s | Tx bytes/s: 53.28 | Tx pkts/s: 1.33

Rx time: 3.00296s | Rx bytes/s: 30.64 | Rx pkts/s: 0.67

Nping done: 2 IP addresses pinged in 4.01 seconds

OPTIONS SUMMARY

This options summary is printed when Nping is run with no arguments. It helps people remember the most common options, but is no substitute for the in-depth documentation in the rest of this manual. Some obscure options aren't even included here.

Nping 0.5.59BETA1 (<http://nmap.org/nping>)

Usage: nping [Probe mode] [Options] {target specification}

TARGET SPECIFICATION:

Targets may be specified as hostnames, IP addresses, networks, etc.

Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0–255.1–254

PROBE MODES:

--tcp-connect : Unprivileged TCP connect probe mode.
 --tcp : TCP probe mode.
 --udp : UDP probe mode.
 --icmp : ICMP probe mode.
 --arp : ARP/RARP probe mode.
 --tr, --traceroute : Traceroute mode (can only be used with TCP/UDP/ICMP modes).

TCP CONNECT MODE:

-p, --dest-port <port spec> : Set destination port(s).
 -g, --source-port <portnumber> : Try to use a custom source port.

TCP PROBE MODE:

-g, --source-port <portnumber> : Set source port.
 -p, --dest-port <port spec> : Set destination port(s).
 --seq <seqnumber> : Set sequence number.
 --flags <flag list> : Set TCP flags (ACK,PSH,RST,SYN,FIN...)
 --ack <acknumber> : Set ACK number.
 --win <size> : Set window size.
 --badsum : Use a random invalid checksum.

UDP PROBE MODE:

-g, --source-port <portnumber> : Set source port.
 -p, --dest-port <port spec> : Set destination port(s).
 --badsum : Use a random invalid checksum.

ICMP PROBE MODE:

--icmp-type <type> : ICMP type.
 --icmp-code <code> : ICMP code.
 --icmp-id <id> : Set identifier.
 --icmp-seq <n> : Set sequence number.
 --icmp-redirect-addr <addr> : Set redirect address.
 --icmp-param-pointer <pnt> : Set parameter problem pointer.
 --icmp-advert-lifetime <time> : Set router advertisement lifetime.
 --icmp-advert-entry <IP,pref> : Add router advertisement entry.
 --icmp-orig-time <timestamp> : Set originate timestamp.
 --icmp-recv-time <timestamp> : Set receive timestamp.
 --icmp-trans-time <timestamp> : Set transmit timestamp.

ARP/RARP PROBE MODE:

--arp-type <type> : Type: ARP, ARP-reply, RARP, RARP-reply.
 --arp-sender-mac <mac> : Set sender MAC address.
 --arp-sender-ip <addr> : Set sender IP address.
 --arp-target-mac <mac> : Set target MAC address.
 --arp-target-ip <addr> : Set target IP address.

IPv4 OPTIONS:

-S, --source-ip : Set source IP address.
 --dest-ip <addr> : Set destination IP address (used as an alternative to {target specification}).
 --tos <tos> : Set type of service field (8bits).
 --id <id> : Set identification field (16 bits).
 --df : Set Don't Fragment flag.
 --mf : Set More Fragments flag.
 --ttl <hops> : Set time to live [0-255].
 --badsum-ip : Use a random invalid checksum.
 --ip-options <S[R [route]]L [route]]T[U ...> : Set IP options
 --ip-options <hex string> : Set IP options
 --mtu <size> : Set MTU. Packets get fragmented if MTU is small enough.

IPv6 OPTIONS:

-6, --IPv6 : Use IP version 6.
 --dest-ip : Set destination IP address (used as an alternative to {target specification}).
 --hop-limit : Set hop limit (same as IPv4 TTL).
 --traffic-class <class> : Set traffic class.
 --flow <label> : Set flow label.

ETHERNET OPTIONS:

--dest-mac <mac> : Set destination mac address. (Disables ARP resolution)
 --source-mac <mac> : Set source MAC address.
 --ether-type <type> : Set EtherType value.

PAYLOAD OPTIONS:

--data <hex string> : Include a custom payload.
 --data-string <text> : Include a custom ASCII text.
 --data-length <len> : Include len random bytes as payload.

ECHO CLIENT/SERVER:

--echo-client <passphrase> : Run Nping in client mode.
 --echo-server <passphrase> : Run Nping in server mode.
 --echo-port <port> : Use custom <port> to listen or connect.
 --no-crypto : Disable encryption and authentication.
 --once : Stop the server after one connection.
 --safe-payloads : Erase application data in echoed packets.

TIMING AND PERFORMANCE:

Options which take <time> are in seconds, or append 'ms' (milliseconds), 's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m, 0.25h).

--delay <time> : Adjust delay between probes.
 --rate <rate> : Send num packets per second.

MISC:

-h, --help : Display help information.
 -V, --version : Display current version number.
 -c, --count <n> : Stop after <n> rounds.
 -e, --interface <name> : Use supplied network interface.
 -H, --hide-sent : Do not display sent packets.
 -N, --no-capture : Do not try to capture replies.
 --privileged : Assume user is fully privileged.
 --unprivileged : Assume user lacks raw socket privileges.
 --send-eth : Send packets at the raw ethernet layer.
 --send-ip : Send packets using raw IP sockets.
 --bpf-filter <filter spec> : Specify custom BPF filter.

OUTPUT:

-v : Increment verbosity level by one.
 -v[level] : Set verbosity level. E.g: -v4
 -d : Increment debugging level by one.
 -d[level] : Set debugging level. E.g: -d3
 -q : Decrease verbosity level by one.
 -q[N] : Decrease verbosity level N times
 --quiet : Set verbosity and debug level to minimum.
 --debug : Set verbosity and debug to the max level.

EXAMPLES:

nping scanme.nmap.org
 nping --tcp -p 80 --flags rst --ttl 2 192.168.1.1
 nping --icmp --icmp-type time --delay 500ms 192.168.254.254
 nping --echo-server "public" -e wlan0 -vvv

```
nping --echo-client "public" echo.nmap.org --tcp -p1-1024 --flags ack
```

SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES

TARGET SPECIFICATION

Everything on the Nping command line that isn't an option or an option argument is treated as a target host specification. Nping uses the same syntax for target specifications that Nmap does. The simplest case is a single target given by IP address or hostname.

Nping supports CIDR-style addressing. You can append */numbits* to an IPv4 address or hostname and Nping will send probes to every IP address for which the first *numbits* are the same as for the reference IP or hostname given. For example, 192.168.10.0/24 would send probes to the 256 hosts between 192.168.10.0 (binary: 11000000 10101000 00001010 00000000) and 192.168.10.255 (binary: 11000000 10101000 00001010 11111111), inclusive. 192.168.10.40/24 would ping exactly the same targets. Given that the host scanme.nmap.org. is at the IP address 64.13.134.52, the specification scanme.nmap.org/16 would send probes to the 65,536 IP addresses between 64.13.0.0 and 64.13.255.255. The smallest allowed value is /0, which targets the whole Internet. The largest value is /32, which targets just the named host or IP address because all address bits are fixed.

CIDR notation is short but not always flexible enough. For example, you might want to send probes to 192.168.0.0/16 but skip any IPs ending with .0 or .255 because they may be used as subnet network and broadcast addresses. Nping supports this through octet range addressing. Rather than specify a normal IP address, you can specify a comma-separated list of numbers or ranges for each octet. For example, 192.168.0-255.1-254 will skip all addresses in the range that end in .0 or .255, and 192.168.3-5.7.1 will target the four addresses 192.168.3.1, 192.168.4.1, 192.168.5.1, and 192.168.7.1. Either side of a range may be omitted; the default values are 0 on the left and 255 on the right. Using - by itself is the same as 0-255, but remember to use 0- in the first octet so the target specification doesn't look like a command-line option. Ranges need not be limited to the final octets: the specifier 0-.-.13.37 will send probes to all IP addresses on the Internet ending in .13.37. This sort of broad sampling can be useful for Internet surveys and research.

IPv6 addresses can only be specified by their fully qualified IPv6 address or hostname. CIDR and octet ranges aren't supported for IPv6 because they are rarely useful.

Nping accepts multiple host specifications on the command line, and they don't need to be the same type. The command **nping scanme.nmap.org 192.168.0.0/8 10.0.0.1,3-7.-** does what you would expect.

OPTION SPECIFICATION

Nping is designed to be very flexible and fit a wide variety of needs. As with most command-line tools, its behavior can be adjusted using command-line options. These general principles apply to option arguments, unless stated otherwise.

Options that take integer numbers can accept values specified in decimal, octal or hexadecimal base. When a number starts with 0x, it will be treated as hexadecimal; when it simply starts with 0, it will be treated as octal. Otherwise, Nping will assume the number has been specified in base 10. Virtually all numbers that can be supplied from the command line are unsigned so, as a general rule, the minimum value is zero.

Users may also specify the word random or rand to make Nping generate a random value within the expected range.

IP addresses may be given as IPv4 addresses (e.g. 192.168.1.1), IPv6 addresses (e.g. 2001:db8:85a3::8e4c:760:7146), or hostnames, which will be resolved using the default DNS server configured in the host system.

Options that take MAC addresses accept the usual colon-separated 6 hex byte format (e.g. 00:50:56:d4:01:98). Hyphens may also be used instead of colons (e.g. 00-50-56-c0-00-08). The special word random or rand sets a random address and the word broadcast or bcast sets ff:ff:ff:ff:ff:ff.

GENERAL OPERATION

Unlike other ping and packet generation tools, Nping supports multiple target host and port specifications. While this provides great flexibility, it is not obvious how Nping handles situations where there is more than one host and/or more than one port to send probes to. This section explains how Nping behaves in these cases.

When multiple target hosts are specified, Nping rotates among them in round-robin fashion. This gives slow hosts more time to send their responses before another probe is sent to them. Ports are also scheduled using round robin. So, unless only one port is specified, Nping never sends two probes to the same target host and port consecutively.

The loop around targets is the “inner loop” and the loop around ports is the “outer loop”. All targets will be sent a probe for a given port before moving on to the next port. Between probes, Nping waits a configurable amount of time called the “inter-probe delay”, which is controlled by the **--delay** option. These examples show how it works.

```
# nping --tcp -c 2 1.1.1.1 -p 100-102
```

```
Starting Nping ( http://nmap.org/nping )
SENT (0.0210s) TCP 192.168.1.77 > 1.1.1.1:100
SENT (1.0230s) TCP 192.168.1.77 > 1.1.1.1:101
SENT (2.0250s) TCP 192.168.1.77 > 1.1.1.1:102
SENT (3.0280s) TCP 192.168.1.77 > 1.1.1.1:100
SENT (4.0300s) TCP 192.168.1.77 > 1.1.1.1:101
SENT (5.0320s) TCP 192.168.1.77 > 1.1.1.1:102
```

```
# nping --tcp -c 2 1.1.1.1 2.2.2.2 3.3.3.3 -p 8080
```

```
Starting Nping ( http://nmap.org/nping )
SENT (0.0230s) TCP 192.168.0.21 > 1.1.1.1:8080
SENT (1.0240s) TCP 192.168.0.21 > 2.2.2.2:8080
SENT (2.0260s) TCP 192.168.0.21 > 3.3.3.3:8080
SENT (3.0270s) TCP 192.168.0.21 > 1.1.1.1:8080
SENT (4.0290s) TCP 192.168.0.21 > 2.2.2.2:8080
SENT (5.0310s) TCP 192.168.0.21 > 3.3.3.3:8080
```

```
# nping --tcp -c 1 --delay 500ms 1.1.1.1 2.2.2.2 3.3.3.3 -p 137-139
```

```
Starting Nping ( http://nmap.org/nping )
SENT (0.0230s) TCP 192.168.0.21 > 1.1.1.1:137
SENT (0.5250s) TCP 192.168.0.21 > 2.2.2.2:137
SENT (1.0250s) TCP 192.168.0.21 > 3.3.3.3:137
SENT (1.5280s) TCP 192.168.0.21 > 1.1.1.1:138
SENT (2.0280s) TCP 192.168.0.21 > 2.2.2.2:138
SENT (2.5310s) TCP 192.168.0.21 > 3.3.3.3:138
SENT (3.0300s) TCP 192.168.0.21 > 1.1.1.1:139
SENT (3.5330s) TCP 192.168.0.21 > 2.2.2.2:139
SENT (4.0330s) TCP 192.168.0.21 > 3.3.3.3:139
```

PROBE MODES

Nping supports a wide variety of protocols. Although in some cases Nping can automatically determine the mode from the options used, it is generally a good idea to specify it explicitly.

--tcp-connect (TCP Connect mode) .

TCP connect mode is the default mode when a user does not have raw packet privileges. Instead of writing raw packets as most other modes do, Nping asks the underlying operating system to establish a connection with the target machine and port by issuing the connect system call. This is the same high-level system call that web browsers, P2P clients, and most other network-enabled applications use to establish a connection. It is part of a programming interface known as the Berkeley Sockets

API. Rather than read raw packet responses off the wire, Nping uses this API to obtain status information on each connection attempt. For this reason, you will not be able to see the contents of the packets that are sent or received but only status information about the TCP connection establishment taking place.

--tcp (TCP mode) .

TCP is the mode that lets users create and send any kind of TCP packet. TCP packets are sent embedded in IP packets that can also be tuned. This mode can be used for many different purposes. For example you could try to discover open ports by sending TCP SYN messages without completing the three-way handshake. This technique is often referred to as half-open scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and then wait for a response. A SYN/ACK indicates the port is open, while a RST indicates it's closed. If no response is received one could assume that some intermediate network device is filtering the responses. Another use could be to see how a remote TCP/IP stack behaves when it receives a non-RFC-compliant packet, like one with both SYN and RST flags set. One could also do some evil by creating custom RST packets using an spoofed IP address with the intent of closing an active TCP connection.

--udp (UDP mode) .

UDP mode can have two different behaviours. Under normal circumstances, it lets users create custom IP/UDP packets. However, if Nping is run by a user without raw packet privileges and no changes to the default protocol headers are requested, then Nping enters the unprivileged UDP mode which basically sends UDP packets to the specified target hosts and ports using the `sendto` system call. Note that in this unprivileged mode it is not possible to see low-level header information of the packets on the wire but only status information about the amount of bytes that are being transmitted and received. UDP mode can be used to interact with any UDP-based server. Examples are DNS servers, streaming servers, online gaming servers, and port knocking/single-packet authorization daemons.

--icmp (ICMP mode) .

ICMP mode is the default mode when the user runs Nping with raw packet privileges. Any kind of ICMP message can be created. The default ICMP type is Echo, i.e., ping. ICMP mode can be used for many different purposes, from a simple request for a timestamp or a netmask to the transmission of fake destination unreachable messages, custom redirects, and router advertisements.

--arp (ARP/RARP mode) .

ARP lets you create and send a few different ARP-related packets. These include ARP, RARP, DRARP, and InARP requests and replies. This mode can be used to perform low-level host discovery, and conduct ARP-cache poisoning attacks.

--traceroute (Traceroute mode) .

Traceroute is not a mode by itself but a complement to TCP, UDP, and ICMP modes. When this option is specified Nping will set the IP TTL value of the first probe to 1. When the next router receives the packet it will drop it due to the expiration of the TTL and it will generate an ICMP destination unreachable message. The next probe will have a TTL of 2 so now the first router will forward the packet while the second router will be the one that drops the packet and generates the ICMP message. The third probe will have a TTL value of 3 and so on. By examining the source addresses of all those ICMP Destination Unreachable messages it is possible to determine the path that the probes take until they reach their final destination.

TCP CONNECT MODE

-p *port_spec*, **--dest-port** *port_spec* (Target ports) .

This option specifies which ports you want to try to connect to. It can be a single port, a comma-separated list of ports (e.g. 80,443,8080), a range (e.g. 1-1023), and any combination of those (e.g. 21-25,80,443,1024-2048). The beginning and/or end values of a range may be omitted, causing Nping to use 1 and 65535, respectively. So you can specify `-p-` to target ports from 1 through 65535. Using port zero is allowed if you specify it explicitly.

-g *portnumber*, **--source-port** *portnumber* (Spoof source port) .

This option asks Nping to use the specified port as source port for the TCP connections. Note that this

might not work on all systems or may require root privileges. Specified value must be an integer in the range [0–65535].

TCP MODE

-p *port_spec*, **--dest-port** *port_spec* (Target ports)

This option specifies which destination ports you want to send probes to. It can be a single port, a comma-separated list of ports (e.g. 80,443,8080), a range (e.g. 1–1023), and any combination of those (e.g. 21–25,80,443,1024–2048). The beginning and/or end values of a range may be omitted, causing Nping to use 1 and 65535, respectively. So you can specify **-p** to target ports from 1 through 65535. Using port zero is allowed if you specify it explicitly.

-g *portnumber*, **--source-port** *portnumber* (Spoof source port)

This option asks Nping to use the specified port as source port for the TCP connections. Note that this might not work on all systems or may require root privileges. Specified value must be an integer in the range [0–65535].

--seq *seqnumber* (Sequence Number) .

Specifies the TCP sequence number. In SYN packets this is the initial sequence number (ISN). In a normal transmission this corresponds to the sequence number of the first byte of data in the segment. *seqnumber* must be a number in the range [0–4294967295].

--flags *flags* (TCP Flags) .

This option specifies which flags should be set in the TCP packet. *flags* may be specified in three different ways:

1. As a comma-separated list of flags, e.g. **--flags syn,ack,rst**
2. As a list of one-character flag initials, e.g. **--flags SAR** tells Nping to set flags SYN, ACK, and RST.
3. As an 8-bit hexadecimal number, where the supplied number is the exact value that will be placed in the flags field of the TCP header. The number should start with the prefix 0x and should be in the range [0x00–0xFF], e.g. **--flags 0x20** sets the URG flag as 0x20 corresponds to binary 00100000 and the URG flag is represented by the third bit.

There are 8 possible flags to set: CWR, ECN, URG, ACK, PSH, RST, SYN, and FIN. The special value ALL means to set all flags. NONE means to set no flags. It is important that if you don't want any flag to be set, you request it explicitly because in some cases the SYN flag may be set by default. Here is a brief description of the meaning of each flag:

CWR (Congestion Window Reduced) .

Set by an ECN-Capable sender when it reduces its congestion window (due to a retransmit timeout, a fast retransmit or in response to an ECN notification).

ECN (Explicit Congestion Notification) .

During the three-way handshake it indicates that sender is capable of performing explicit congestion notification. Normally it means that a packet with the IP Congestion Experienced flag set was received during normal transmission. See RFC 3168. for more information.

URG (Urgent) .

Segment is urgent and the urgent pointer field carries valid information.

ACK (Acknowledgement) .

The segment carries an acknowledgement and the value of the acknowledgement number field is valid and contains the next sequence number that is expected from the receiver.

PSH (Push) .

The data in this segment should be immediately pushed to the application layer on arrival.

RST (Reset) .

There was some problem and the sender wants to abort the connection.

SYN (Synchronize) .

The segment is a request to synchronize sequence numbers and establish a connection. The sequence number field contains the sender's initial sequence number.

FIN (Finish) .

The sender wants to close the connection.

--win size (Window Size) .

Specifies the TCP window size, this is, the number of octets the sender of the segment is willing to accept from the receiver at one time. This is usually the size of the reception buffer that the OS allocates for a given connection. *size* must be a number in the range [0–65535].

--badsum (Invalid Checksum) .

Asks Nping to use an invalid TCP checksum for the packets sent to target hosts. Since virtually all host IP stacks properly drop these packets, any responses received are likely coming from a firewall or an IDS that didn't bother to verify the checksum. For more details on this technique, see

<http://nmap.org/p60-12.html>.

UDP MODE

--p port_spec, --dest-port port_spec (Target ports) .

This option specifies which ports you want UDP datagrams to be sent to. It can be a single port, a comma-separated list of ports (e.g. 80,443,8080), a range (e.g. 1–1023), and any combination of those (e.g. 21–25,80,443,1024–2048). The beginning and/or end values of a range may be omitted, causing Nping to use 1 and 65535, respectively. So you can specify **--p** to target ports from 1 through 65535. Using port zero is allowed if you specify it explicitly.

--g portnumber, --source-port portnumber (Spoof source port) .

This option asks Nping to use the specified port as source port for the transmitted datagrams. Note that this might not work on all systems or may require root privileges. Specified value must be an integer in the range [0–65535].

--badsum (Invalid Checksum)

Asks Nping to use an invalid UDP checksum for the packets sent to target hosts. Since virtually all host IP stacks properly drop these packets, any responses received are likely coming from a firewall or an IDS that didn't bother to verify the checksum. For more details on this technique, see

<http://nmap.org/p60-12.html>.

ICMP MODE

--icmp-type type (ICMP type) .

This option specifies which type of ICMP messages should be generated. *type* can be supplied in two different ways. You can use the [official type numbers assigned by IANA](#)^[1] (e.g. **--icmp-type 8** for ICMP Echo Request), or you can use any of the mnemonics listed in the section called “ICMP Types”.

--icmp-code code (ICMP code) .

This option specifies which ICMP code should be included in the generated ICMP messages. *code* can be supplied in two different ways. You can use the [official code numbers assigned by IANA](#)^[1] (e.g. **--icmp-code 1** for Fragment Reassembly Time Exceeded), or you can use any of the mnemonics listed in the section called “ICMP Codes”.

--icmp-id id (ICMP identifier) .

This option specifies the value of the identifier used in some of the ICMP messages. In general it is used to match request and reply messages. *id* must be a number in the range [0–65535].

--icmp-seq seq (ICMP sequence) .

This option specifies the value of the sequence number field used in some ICMP messages. In general it is used to match request and reply messages. *id* must be a number in the range [0–65535].

--icmp-redirect-addr addr (ICMP Redirect address) .

This option sets the address field in ICMP Redirect messages. In other words, it sets the IP address of the router that should be used when sending IP datagrams to the original destination. *addr* can be either an IPv4 address or a hostname.

--icmp-param-pointer pointer (ICMP Parameter Problem pointer) .

This option specifies the pointer that indicates the location of the problem in ICMP Parameter Problem messages. *pointer* should be a number in the range [0–255]. Normally this option is only used when ICMP code is set to 0 ("Pointer indicates the error").

--icmp-advert-lifetime *ttl* (ICMP Router Advertisement Lifetime) .

This option specifies the router advertisement lifetime, this is, the number of seconds the information carried in an ICMP Router Advertisement can be considered valid for. *ttl* must be a positive integer in the range [0–65535].

--icmp-advert-entry *addr,pref* (ICMP Router Advertisement Entry) .

This option adds a Router Advertisement entry to an ICMP Router Advertisement message. The parameter must be two values separated by a comma. *addr* is the router's IP and can be specified either as an IP address in dot-decimal notation or as a hostname. *pref* is the preference level for the specified IP. It must be a number in the range [0–4294967295]. An example is **--icmp-advert-entry 192.168.128.1,3**.

--icmp-orig-time *timestamp* (ICMP Originate Timestamp) .

This option sets the Originate Timestamp in ICMP Timestamp messages. The Originate Timestamp is expressed as the number of milliseconds since midnight UTC and it corresponds to the time the sender last touched the Timestamp message before its transmission. *timestamp* can be specified as a regular time (e.g. 10s, 3h, 1000ms), or the special string now. You can add or subtract values from now, for example **--icmp-orig-time now-2s**, **--icmp-orig-time now+1h**, **--icmp-orig-time now+200ms**.

--icmp-recv-time *timestamp* (ICMP Receive Timestamp) .

This option sets the Receive Timestamp in ICMP Timestamp messages. The Receive Timestamp is expressed as the number of milliseconds since midnight UTC and it corresponds to the time the echoer first touched the Timestamp message on receipt. *timestamp* is as with **--icmp-orig-time**.

--icmp-trans-time *timestamp* (ICMP Transmit Timestamp) .

This option sets the Transmit Timestamp in ICMP Timestamp messages. The Transmit Timestamp is expressed as the number of milliseconds since midnight UTC and it corresponds to the time the echoer last touched the Timestamp message before its transmission. *timestamp* is as with **--icmp-orig-time**.

ICMP Types

These identifiers may be used as mnemonics for the ICMP type numbers given to the **--icmp-type** option. In general there are three forms of each identifier: the full name (e.g. destination-unreachable), the short name (e.g. dest-unr), or the initials (e.g. du). In ICMP types that request something, the word "request" is omitted.

echo-reply, echo-rep, er

Echo Reply (type 0). This message is sent in response to an Echo Request message.

destination-unreachable, dest-unr, du

Destination Unreachable (type 3). This message indicates that a datagram could not be delivered to its destination.

source-quench, sour-que, sq

Source Quench (type 4). This message is used by a congested IP device to tell other device that is sending packets too fast and that it should slow down.

redirect, redi, r

Redirect (type 5). This message is normally used by routers to inform a host that there is a better route to use for sending datagrams. See also the **--icmp-redirect-addr** option.

echo-request, echo, e

Echo Request (type 8). This message is used to test the connectivity of another device on a network.

router-advertisement, rout-adv, ra

Router Advertisement (type 9). This message is used by routers to let hosts know of their existence and capabilities. See also the **--icmp-advert-lifetime** option.

router-solicitation, rout-sol, rs

Router Solicitation (type 10). This message is used by hosts to request Router Advertisement messages from any listening routers.

time-exceeded, time-exc, te

Time Exceeded (type 11). This message is generated by some intermediate device (normally a router) to indicate that a datagram has been discarded before reaching its destination because the IP TTL expired.

parameter-problem, member-pro, pp

Parameter Problem (type 12). This message is used when a device finds a problem with a parameter in an IP header and it cannot continue processing it. See also the **--icmp-param-pointer** option.

timestamp, time, tm

Timestamp Request (type 13). This message is used to request a device to send a timestamp value for propagation time calculation and clock synchronization. See also the **--icmp-orig-time**, **--icmp-recv-time**, and **--icmp-trans-time**.

timestamp-reply, time-rep, tr

Timestamp Reply (type 14). This message is sent in response to a Timestamp Request message.

information, info, i

Information Request (type 15). This message is now obsolete but it was originally used to request configuration information from another device.

information-reply, info-rep, ir

Information Reply (type 16). This message is now obsolete but it was originally sent in response to an Information Request message to provide configuration information.

mask-request, mask, m

Address Mask Request (type 17). This message is used to ask a device to send its subnet mask.

mask-reply, mask-rep, mr

Address Mask Reply (type 18). This message contains a subnet mask and is sent in response to a Address Mask Request message.

traceroute, trace, tc

Traceroute (type 30). This message is normally sent by an intermediate device when it receives an IP datagram with a traceroute option. ICMP Traceroute messages are still experimental, see RFC 1393. for more information.

ICMP Codes

These identifiers may be used as mnemonics for the ICMP code numbers given to the **--icmp-code** option. They are listed by the ICMP type they correspond to.

Destination Unreachable

network-unreachable, netw-unr, net

Code 0. Datagram could not be delivered to its destination network (probably due to some routing problem).

host-unreachable, host-unr, host

Code 1. Datagram was delivered to the destination network but it was impossible to reach the specified host (probably due to some routing problem).

protocol-unreachable, prot-unr, proto

Code 2. The protocol specified in the Protocol field of the IP datagram is not supported by the host to which the datagram was delivered.

port-unreachable, port-unr, port

Code 3. The TCP/UDP destination port was invalid.

needs-fragmentation, need-fra, frag

Code 4. Datagram had the DF bit set but it was too large for the MTU of the next physical

network so it had to be dropped.

source-route-failed, sour-rou, routefail

Code 5. IP datagram had a Source Route option but a router couldn't pass it to the next hop.

network-unknown, netw-unk, net?

Code 6. Destination network is unknown. This code is never used. Instead, Network Unreachable is used.

host-unknown, host-unk, host?

Code 7. Specified host is unknown. Usually generated by a router local to the destination host to inform of a bad address.

host-isolated, host-iso, isolated

Code 8. Source Host Isolated. Not used.

network-prohibited, netw-pro, !net

Code 9. Communication with destination network is administratively prohibited (source device is not allowed to send packets to the destination network).

host-prohibited, host-pro, !host

Code 10. Communication with destination host is administratively prohibited. (The source device is allowed to send packets to the destination network but not to the destination device.)

network-tos, unreachable-network-tos, netw-tos, tosnet

Code 11. Destination network unreachable because it cannot provide the type of service specified in the IP TOS field.

host-tos, unreachable-host-tos, toshost

Code 12. Destination host unreachable because it cannot provide the type of service specified in the IP TOS field.

communication-prohibited, comm-pro, !comm

Code 13. Datagram could not be forwarded due to filtering that blocks the message based on its contents.

host-precedence-violation, precedence-violation, prec-vio, violation

Code 14. Precedence value in the IP TOS field is not permitted.

precedence-cutoff, prec-cut, cutoff

Code 15. Precedence value in the IP TOS field is lower than the minimum allowed for the network.

Redirect

redirect-network, redi-net, net

Code 0. Redirect all future datagrams with the same destination network as the original datagram, to the router specified in the Address field. The use of this code is prohibited by RFC 1812..

redirect-host, redi-host, host

Code 1. Redirect all future datagrams with the same destination host as the original datagram, to the router specified in the Address field.

redirect-network-tos, redi-ntos, redir-ntos

Code 2. Redirect all future datagrams with the same destination network and IP TOS value as the original datagram, to the router specified in the Address field. The use of this code is prohibited by RFC 1812.

redirect-host-tos, redi-htos, redir-htos

Code 3. Redirect all future datagrams with the same destination host and IP TOS value as the original datagram, to the router specified in the Address field.

Router Advertisement

normal-advertisement, norm-adv, normal, zero, default, def

Code 0. Normal router advertisement. In Mobile IP: Mobility agent can act as a router for IP datagrams not related to mobile nodes.

not-route-common-traffic, not-rou, mobile-ip, !route, !commontraffic

Code 16. Used for Mobile IP. The mobility agent does not route common traffic. All foreign agents must forward to a default router any datagrams received from a registered mobile node

Time Exceeded

ttl-exceeded-in-transit, ttl-exc, ttl-transit

Code 0. IP Time To Live expired during transit.

fragment-reassembly-time-exceeded, frag-exc, frag-time

Code 1. Fragment reassembly time has been exceeded.

Parameter Problem

pointer-indicates-error, poin-ind, pointer

Code 0. The pointer field indicates the location of the problem. See the **--icmp-param-pointer** option.

missing-required-option, miss-option, option-missing

Code 1. IP datagram was expected to have an option that is not present.

bad-length, bad-len, badlen

Code 2. The length of the IP datagram is incorrect.

ARP MODE

--arp-type *type* (ICMP Type) .

This option specifies which type of ARP messages should be generated. *type* can be supplied in two different ways. You can use the [official numbers assigned by IANA](#)^[2] (e.g. **--arp-type 1** for ARP Request), or you can use one of the mnemonics from the section called “ARP Types”.

--arp-sender-mac *mac* (Sender MAC address) .

This option sets the Sender Hardware Address field of the ARP header. Although ARP supports many types of link layer addresses, currently Nping only supports MAC addresses. *mac* must be specified using the traditional MAC notation (e.g. 00:0a:8a:32:f4:ae). You can also use hyphens as separators (e.g. 00-0a-8a-32-f4-ae).

--arp-sender-ip *addr* (Sender IP address) .

This option sets the Sender IP field of the ARP header. *addr* can be given as an IPv4 address or a hostname.

--arp-target-mac *mac* (target MAC address) .

This option sets the Target Hardware Address field of the ARP header.

--arp-target-ip *addr* (target ip address) .

This option sets the Target IP field of the ARP header.

ARP Types

These identifiers may be used as mnemonics for the ARP type numbers given to the **--arp-type** option.

arp-request, arp, a

ARP Request (type 1). ARP requests are used to translate network layer addresses (normally IP addresses) to link layer addresses (usually MAC addresses). Basically, an ARP request is a broadcasted message that asks the host in the same network segment that has a given IP address to provide its MAC address.

arp-reply, arp-rep, ar

ARP Reply (type 2). An ARP reply is a message that a host sends in response to an ARP request to provide its link layer address.

rarp-request, rarp, r

RARP Requests (type 3). RARP requests are used to translate a link layer address (normally a MAC

address) to a network layer address (usually an IP address). Basically a RARP request is a broadcasted message sent by a host that wants to know his own IP address because it doesn't have any. It was the first protocol designed to solve the bootstrapping problem. However, RARP is now obsolete and DHCP is used instead. For more information about RARP see RFC 903..

rarp-reply, rarp-rep, rr

RARP Reply (type 4). A RARP reply is a message sent in response to a RARP request to provide an IP address to the host that sent the RARP request in the first place.

drarp-request, drarp, d

Dynamic RARP Request (type 5). Dynamic RARP is an extension to RARP used to obtain or assign a network layer address from a fixed link layer address. DRARP was used mainly in Sun Microsystems platforms in the late 90's but now it's no longer used. See RFC 1931. for more information.

drarp-reply, drarp-rep, dr

Dynamic RARP Reply (type 6). A DRARP reply is a message sent in response to a RARP request to provide network layer address.

drarp-error, drarp-err, de

DRARP Error (type 7). DRARP Error messages are usually sent in response to DRARP requests to inform of some error. In DRARP Error messages, the Target Protocol Address field is used to carry an error code (usually in the first byte). The error code is intended to tell why no target protocol address is being returned. For more information see RFC 1931.

inarp-request, inarp, i

Inverse ARP Request (type 8). InARP requests are used to translate a link layer address to a network layer address. It is similar to RARP request but in this case, the sender of the InARP request wants to know the network layer address of another node, not its own address. InARP is mainly used in Frame Relay and ATM networks. For more information see RFC 2390..

inarp-reply, inarp-rep, ir

Inverse ARP Reply (type 9). InARP reply messages are sent in response to InARP requests to provide the network layer address associated with the host that has a given link layer address.

arp-nak, an

ARP NAK (type 10). ARP NAK messages are an extension to the ATMARF protocol and they are used to improve the robustness of the ATMARF server mechanism. With ARP NAK, a client can determine the difference between a catastrophic server failure and an ATMARF table lookup failure. See RFC 1577. for more information.

IPV4 OPTIONS

-S *addr*, --source-ip *addr* (Source IP Address) .

Sets the source IP address. This option lets you specify a custom IP address to be used as source IP address in sent packets. This allows spoofing the sender of the packets. *addr* can be an IPv4 address or a hostname.

--dest-ip *addr* (Destination IP Address) .

Adds a target to Nping's target list. This option is provided for consistency but its use is deprecated in favor of plain target specifications. See the section called "TARGET SPECIFICATION".

--tos *tos* (Type of Service) .

Sets the IP TOS field. The TOS field is used to carry information to provide quality of service features. It is normally used to support a technique called Differentiated Services. See RFC 2474. for more information. *tos* must be a number in the range [0–255].

--id *id* (Identification) .

Sets the IPv4 Identification field. The Identification field is a 16-bit value that is common to all fragments belonging to a particular message. The value is used by the receiver to reassemble the original message from the fragments received. *id* must be a number in the range [0–65535].

--df (Don't Fragment) .

Sets the Don't Fragment bit in sent packets. When an IP datagram has its DF flag set, intermediate

devices are not allowed to fragment it so if it needs to travel across a network with a MTU smaller than datagram length the datagram will have to be dropped. Normally an ICMP Destination Unreachable message is generated and sent back to the sender.

--md (More Fragments) .

Sets the More Fragments bit in sent packets. The MF flag is set to indicate the receiver that the current datagram is a fragment of some larger datagram. When set to zero it indicates that the current datagram is either the last fragment in the set or that it is the only fragment.

--ttl *hops* (Time To Live) .

Sets the IPv4 Time-To-Live (TTL) field in sent packets to the given value. The TTL field specifies how long the datagram is allowed to exist on the network. It was originally intended to represent a number of seconds but it actually represents the number of hops a packet can traverse before being dropped. The TTL tries to avoid a situation in which undeliverable datagrams keep being forwarded from one router to another endlessly. *hops* must be a number in the range [0–255].

--badsum-ip (Invalid IP checksum) .

Asks Nping to use an invalid IP checksum for packets sent to target hosts. Note that some systems (like most Linux kernels), may fix the checksum before placing the packet on the wire, so even if Nping shows the incorrect checksum in its output, the packets may be transparently corrected by the kernel.

--ip-options *S*[*R* [*route*]]*L* [*route*]]*T**U* ..., **--ip-options** *hex string* (IP Options) .

The IP protocol offers several options which may be placed in packet headers. Unlike the ubiquitous TCP options, IP options are rarely seen due to practicality and security concerns. In fact, many Internet routers block the most dangerous options such as source routing. Yet options can still be useful in some cases for determining and manipulating the network route to target machines. For example, you may be able to use the record route option to determine a path to a target even when more traditional traceroute-style approaches fail. Or if your packets are being dropped by a certain firewall, you may be able to specify a different route with the strict or loose source routing options.

The most powerful way to specify IP options is to simply pass in hexadecimal data as the argument to **--ip-options**. Precede each hex byte value with \x. You may repeat certain characters by following them with an asterisk and then the number of times you wish them to repeat. For example, \x01\x07\x04\x00*4 is the same as \x01\x07\x04\x00\x00\x00\x00.

Note that if you specify a number of bytes that is not a multiple of four, an incorrect IP header length will be set in the IP packet. The reason for this is that the IP header length field can only express multiples of four. In those cases, the length is computed by dividing the header length by 4 and rounding down. This will affect the way the header that follows the IP header is interpreted, showing bogus information in Nping or in the output of any sniffer. Although this kind of situation might be useful for some stack stress tests, users would normally want to specify explicit padding, so the correct header length is set.

Nping also offers a shortcut mechanism for specifying options. Simply pass the letter R, T, or U to request record-route, record-timestamp, or both options together, respectively. Loose or strict source routing may be specified with an L or S followed by a space and then a space-separated list of IP addresses.

For more information and examples of using IP options with Nping, see the mailing list post at <http://seclists.org/nmap-dev/2006/q3/0052.html>.

--mtu *size* (Maximum Transmission Unit) .

This option sets a fictional MTU in Nping so IP datagrams larger than *size* are fragmented before transmission. *size* must be specified in bytes and corresponds to the number of octets that can be carried on a single link-layer frame.

IPV6 OPTIONS

-6, --ipv6 (Use IPv6) .

Tells Nping to use IP version 6 instead of the default IPv4. It is generally a good idea to specify this option as early as possible in the command line so Nping can parse it soon and know in advance that the rest of the parameters refer to IPv6. The command syntax is the same as usual except that you also add the **-6** option. Of course, you must use IPv6 syntax if you specify an address rather than a hostname. An address might look like **3ffe:7501:4819:2000:210:f3ff:fe03:14d0**, so hostnames are recommended.

While IPv6 hasn't exactly taken the world by storm, it gets significant use in some (usually Asian) countries and most modern operating systems support it. To use Nping with IPv6, both the source and target of your packets must be configured for IPv6. If your ISP (like most of them) does not allocate IPv6 addresses to you, free tunnel brokers are widely available and work fine with Nping. You can use the free IPv6 tunnel broker service at <http://www.tunnelbroker.net>.

Please note that IPv6 support is still highly experimental and many modes and options may not work with it.

-S addr, --source-ip addr (Source IP Address) .

Sets the source IP address. This option lets you specify a custom IP address to be used as source IP address in sent packets. This allows spoofing the sender of the packets. *addr* can be an IPv6 address or a hostname.

--dest-ip addr (Destination IP Address) .

Adds a target to Nping's target list. This option is provided for consistency but its use is deprecated in favor of plain target specifications. See the section called "TARGET SPECIFICATION".

--flow label (Flow Label) .

Sets the IPv6 Flow Label. The Flow Label field is 20 bits long and is intended to provide certain quality-of-service properties for real-time datagram delivery. However, it has not been widely adopted, and not all routers or endpoints support it. Check RFC 2460. for more information. *label* must be an integer in the range [0-1048575].

--traffic-class class (Traffic Class) .

Sets the IPv6 Traffic Class. This field is similar to the TOS field in IPv4, and is intended to provide the Differentiated Services method, enabling scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. Check RFC 2474. for more information. *class* must be an integer in the range [0-255].

--hop-limit hops (Hop Limit) .

Sets the IPv6 Hop Limit field in sent packets to the given value. The Hop Limit field specifies how long the datagram is allowed to exist on the network. It represents the number of hops a packet can traverse before being dropped. As with the TTL in IPv4, IPv6 Hop Limit tries to avoid a situation in which undeliverable datagrams keep being forwarded from one router to another endlessly. *hops* must be a number in the range [0-255].

ETHERNET OPTIONS

In most cases Nping sends packets at the raw IP level. This means that Nping creates its own IP packets and transmits them through a raw socket. However, in some cases it may be necessary to send packets at the raw Ethernet level. This happens, for example, when Nping is run under Windows (as Microsoft has disabled raw socket support since Windows XP SP2), or when Nping is asked to send ARP packets. Since in some cases it is necessary to construct ethernet frames, Nping offers some options to manipulate the different fields.

--dest-mac mac (Ethernet Destination MAC Address) .

This option sets the destination MAC address that should be set in outgoing Ethernet frames. This is useful in case Nping can't determine the next hop's MAC address or when you want to route probes through a router other than the configured default gateway. The MAC address should have the usual format of six colon-separated bytes, e.g. 00:50:56:d4:01:98. Alternatively, hyphens may be used

instead of colons. Use the word random or rand to generate a random address, and broadcast or bcast to use ff:ff:ff:ff:ff:ff. If you set up a bogus destination MAC address your probes may not reach the intended targets.

--source-mac *mac* (Ethernet Source MAC Address) .

This option sets the source MAC address that should be set in outgoing Ethernet frames. This is useful in case Nping can't determine your network interface MAC address or when you want to inject traffic into the network while hiding your network card's real address. The syntax is the same as for **--dest-mac**. If you set up a bogus source MAC address you may not receive probe replies.

--ether-type *type* (Ethertype) .

This option sets the Ethertype field of the ethernet frame. The Ethertype is used to indicate which protocol is encapsulated in the payload. *type* can be supplied in two different ways. You can use the [official numbers listed by the IEEE](#)^[3] (e.g. **--ether-type 0x0800** for IP version 4), or one of the mnemonics from the section called "Ethernet Types".

Ethernet Types

These identifiers may be used as mnemonics for the Ethertype numbers given to the **--arp-type** option.

ipv4, ip, 4

Internet Protocol version 4 (type 0x0800).

ipv6, 6

Internet Protocol version 6 (type 0x86DD).

arp

Address Resolution Protocol (type 0x0806).

rarp

Reverse Address Resolution Protocol (type 0x8035).

frame-relay, frelay, fr

Frame Relay (type 0x0808).

ppp

Point-to-Point Protocol (type 0x880B).

gsmpp

General Switch Management Protocol (type 0x880C).

mpls

Multiprotocol Label Switching (type 0x8847).

mps-ual, mps

Multiprotocol Label Switching with Upstream-assigned Label (type 0x8848).

mcap

Multicast Channel Allocation Protocol (type 0x8861).

pppoe-discovery, pppoe-d

PPP over Ethernet Discovery Stage (type 0x8863).

pppoe-session, pppoe-s

PPP over Ethernet Session Stage (type 0x8864).

ctag

Customer VLAN Tag Type (type 0x8100).

epon

Ethernet Passive Optical Network (type 0x8808).

pbnac

Port-based network access control (type 0x888E).

stag

Service VLAN tag identifier (type 0x88A8).

ethexp1
Local Experimental Ethertype 1 (type 0x88B5).

ethexp2
Local Experimental Ethertype 2 (type 0x88B6).

ethoui
OUI Extended Ethertype (type 0x88B7).

preauth
Pre-Authentication (type 0x88C7).

lldp
Link Layer Discovery Protocol (type 0x88CC).

mac-security, mac-sec, macsec
Media Access Control Security (type 0x88E5).

mvrp
Multiple VLAN Registration Protocol (type 0x88F5).

mmrp
Multiple Multicast Registration Protocol (type 0x88F6).

frrr
Fast Roaming Remote Request (type 0x890D).

PAYLOAD OPTIONS

--data *hex string* (Append custom binary data to sent packets) .
This option lets you include binary data as payload in sent packets. *hex string* may be specified in any of the following formats: 0xAABBCCDDEEFF..., AABBCCDDEEFF... or \xAA\xBB\xCC\xDD\xEE\xFF.... Examples of use are **--data 0xdeadbeef** and **--data \xCA\xFE\x09**. Note that if you specify a number like 0x00ff no byte-order conversion is performed. Make sure you specify the information in the byte order expected by the receiver.

--data-string *string* (Append custom string to sent packets) .
This option lets you include a regular string as payload in sent packets. *string* can contain any string. However, note that some characters may depend on your system's locale and the receiver may not see the same information. Also, make sure you enclose the string in double quotes and escape any special characters from the shell. Example: **--data-string "Jimmy Jazz..."**.

--data-length *len* (Append random data to sent packets) .
This option lets you include *len* random bytes of data as payload in sent packets. *len* must be an integer in the range [0-65400]. However, values higher than 1400 are not recommended because it may not be possible to transmit packets due to network MTU limitations.

ECHO MODE

The "Echo Mode" is a novel technique implemented by Nping which lets users see how network packets change in transit, from the host where they originated to the target machine. Basically, the Echo mode turns Nping into two different pieces: the Echo server and the Echo client. The Echo server is a network service that has the ability to capture packets from the network and send a copy ("echo them") to the originating client through a side TCP channel. The Echo client is the part that generates such network packets, transmits them to the server, and receives their echoed version through a side TCP channel that it has previously established with the Echo server.

This scheme lets the client see the differences between the packets that it sends and what is actually received by the server. By having the server send back copies of the received packets through the side channel, things like NAT devices become immediately apparent to the client because it notices the changes in the source IP address (and maybe even source port). Other devices like those that perform traffic shaping, changing TCP window sizes or adding TCP options transparently between hosts, turn up too.

The Echo mode is also useful for troubleshooting routing and firewall issues. Among other things, it can be used to determine if the traffic generated by the Nping client is being dropped in transit and never gets to its

destination or if the responses are the ones that don't get back to it.

Internally, client and server communicate over an encrypted and authenticated channel, using the Nping Echo Protocol (NEP), whose technical specification can be found in <http://nmap.org/svn/nping/docs/EchoProtoRFC.txt>

The following paragraphs describe the different options available in Nping's Echo mode.

--ec *passphrase*, --echo-client *passphrase* (Run Echo client) .

This option tells Nping to run as an Echo client. *passphrase* is a sequence of ASCII characters that is used to generate the cryptographic keys needed for encryption and authentication in a given session. The passphrase should be a secret that is also known by the server, and it may contain any number of printable ASCII characters. Passphrases that contain whitespace or special characters must be enclosed in double quotes.

When running Nping as an Echo client, most options from the regular raw probe modes apply. The client may be configured to send specific probes using flags like **--tcp**, **--icmp** or **--udp**. Protocol header fields may be manipulated normally using the appropriate options (e.g. **--ttl**, **--seq**, **--icmp-type**, etc.). The only exceptions are ARP-related flags, which are not supported in Echo mode, as protocols like ARP are closely related to the data link layer and its probes can't pass through different network segments.

--es *passphrase*, --echo-server *passphrase* (Run Echo server) .

This option tells Nping to run as an Echo server. *passphrase* is a sequence of ASCII characters that is used to generate the cryptographic keys needed for encryption and authentication in a given session. The passphrase should be a secret that is also known by the clients, and it may contain any number of printable ASCII characters. Passphrases that contain whitespace or special characters must be enclosed in double quotes. Note that although it is not recommended, it is possible to use empty passphrases, supplying **--echo-server ""**. However, if what you want is to set up an open Echo server, it is better to use option **--no-crypto**. See below for details.

--ep *port*, --echo-port *port* (Set Echo TCP port number) .

This option asks Nping to use the specified TCP port number for the Echo side channel connection. If this option is used with **--echo-server**, it specifies the port on which the server listens for connections. If it is used with **--echo-client**, it specifies the port to connect to on the remote host. By default, port number 9929 is used.

--nc, --no-crypto (Disable encryption and authentication) .

This option asks Nping not to use any cryptographic operations during an Echo session. In practical terms, this means that the Echo side channel session data will be transmitted in the clear, and no authentication will be performed by the server or client during the session establishment phase. When **--no-crypto** is used, the passphrase supplied with **--echo-server** or **--echo-client** is ignored.

This option must be specified if Nping was compiled without openssl support. Note that, for technical reasons, a passphrase still needs to be supplied after the **--echo-client** or **--echo-server** flags, even though it will be ignored.

The **--no-crypto** flag might be useful when setting up a public Echo server, because it allows users to connect to the Echo without the need for any passphrase or shared secret. However, it is strongly recommended to not use **--no-crypto** unless absolutely necessary. Public Echo servers should be configured to use the passphrase "public" or the empty passphrase (**--echo-server ""**) as the use of cryptography does not only provide confidentiality and authentication but also message integrity.

--once (Serve one client and quit) .

This option asks the Echo server to quit after serving one client. This is useful when only a single Echo session wants to be established as it eliminates the need to access the remote host to shutdown the server.

--safe-payloads (Zero application data before echoing a packet) .

This option asks the Echo server to erase any application layer data found in client packets before echoing them. When the option is enabled, the Echo server parses the packets received from Echo clients and tries to determine if they contain data beyond the transport layer. If such data is found, it is overwritten with zeroes before transmitting the packets to the appropriate Echo client.

Echo servers can handle multiple simultaneous clients running multiple echo sessions in parallel. In order to determine which packet needs to be echoed to which client and through which session, the Echo server uses an heuristic algorithm. Although we have taken every security measure that we could think of to prevent that a client receives an echoed packet that it did not generate, there is always a risk that our algorithm makes a mistake and delivers a packet to the wrong client. The **--safe-payloads** option is useful for public echo servers or critical deployments where that kind of mistake cannot be afforded.

The following examples illustrate how Nping's Echo mode can be used to discover intermediate devices.

Example 2. Discovering NAT devices

```
# nping --echo-client "public" echo.nmap.org --udp
```

Starting Nping (<http://nmap.org/nping>)

```
SENT (1.0970s) UDP 10.1.20.128:53 > 178.79.165.17:40125 ttl=64 id=32523 iplen=28
CAPT (1.1270s) UDP 80.38.10.21:45657 > 178.79.165.17:40125 ttl=54 id=32523 iplen=28
RCVD (1.1570s) ICMP 178.79.165.17 > 10.1.20.128 Port unreachable (type=3/code=3) ttl=49 id=16619 iplen=56
[...]
SENT (5.1020s) UDP 10.1.20.128:53 > 178.79.165.17:40125 ttl=64 id=32523 iplen=28
CAPT (5.1335s) UDP 80.38.10.21:45657 > 178.79.165.17:40125 ttl=54 id=32523 iplen=28
RCVD (5.1600s) ICMP 178.79.165.17 > 10.1.20.128 Port unreachable (type=3/code=3) ttl=49 id=16623 iplen=56
```

```
Max rtt: 60.628ms | Min rtt: 58.378ms | Avg rtt: 59.389ms
Raw packets sent: 5 (140B) | Rcvd: 5 (280B) | Lost: 0 (0.00%) | Echoed: 5 (140B)
Tx time: 4.00459s | Tx bytes/s: 34.96 | Tx pkts/s: 1.25
Rx time: 5.00629s | Rx bytes/s: 55.93 | Rx pkts/s: 1.00
Nping done: 1 IP address pinged in 6.18 seconds
```

The output clearly shows the presence of a NAT device in the client's local network. Note how the captured packet (CAPT) differs from the SENT packet: the source address for the original packets is in the reserved 10.0.0.0/8 range, while the address seen by the server is 80.38.10.21, the Internet side address of the NAT device. The source port was also modified by the device. The line starting with RCVD corresponds to the responses generated by the TCP/IP stack of the machine where the Echo server is run.

Example 3. Discovering a transparent proxy

```
# nping --echo-client "public" echo.nmap.org --tcp -p80
```

Starting Nping (<http://nmap.org/nping>)

```
SENT (1.2160s) TCP 10.0.1.77:41659 > 178.79.165.17:80 S ttl=64 id=3317 iplen=40 seq=567704200 win=1480
RCVD (1.2180s) TCP 178.79.165.17:80 > 10.0.1.77:41659 SA ttl=128 id=13177 iplen=44 seq=3647106954 win=163;
SENT (2.2150s) TCP 10.0.1.77:41659 > 178.79.165.17:80 S ttl=64 id=3317 iplen=40 seq=567704200 win=1480
SENT (3.2180s) TCP 10.0.1.77:41659 > 178.79.165.17:80 S ttl=64 id=3317 iplen=40 seq=567704200 win=1480
SENT (4.2190s) TCP 10.0.1.77:41659 > 178.79.165.17:80 S ttl=64 id=3317 iplen=40 seq=567704200 win=1480
SENT (5.2200s) TCP 10.0.1.77:41659 > 178.79.165.17:80 S ttl=64 id=3317 iplen=40 seq=567704200 win=1480
```

```
Max rtt: 2.062ms | Min rtt: 2.062ms | Avg rtt: 2.062ms
Raw packets sent: 5 (200B) | Rcvd: 1 (46B) | Lost: 4 (80.00%) | Echoed: 0 (0B)
Tx time: 4.00504s | Tx bytes/s: 49.94 | Tx pkts/s: 1.25
```

```
Rx time: 5.00618s | Rx bytes/s: 9.19 | Rx pkts/s: 0.20
Nping done: 1 IP address pinged in 6.39 seconds
```

In this example, the output is a bit more tricky. The absence of error messages shows that the Echo client has successfully established an Echo session with the server. However, no CAPT packets can be seen in the output. This means that none of the transmitted packets reached the server. Interestingly, a TCP SYN-ACK packet was received in response to the first TCP-SYN packet (and also, it is known that the target host does not have port 80 open). This behavior reveals the presence of a transparent web proxy cache server (which in this case is an old MS ISA server).

TIMING AND PERFORMANCE OPTIONS

--delay *time* (Delay between probes) .

This option lets you control for how long will Nping wait before sending the next probe. Like in many other ping tools, the default delay is one second. *time* must be a positive integer or floating point number. By default it is specified in seconds, however you can give an explicit unit by appending ms for milliseconds, s for seconds, m for minutes, or h for hours (e.g. 2.5s, 45m, 2h).

--rate *rate* (Send probes at a given rate) .

This option specifies the number of probes that Nping should send per second. This option and **--delay** are inverses; **--rate 20** is the same as **--delay 0.05**. If both options are used, only the last one in the parameter list counts.

MISCELLANEOUS OPTIONS

-h, --help (Display help) .

Displays help information and exits.

-V, --version (Display version) .

Displays the program's version number and quits.

-c *rounds*, **--count** *rounds* (Stop after a given number of rounds) .

This option lets you specify the number of times that Nping should loop over target hosts (and in some cases target ports). Nping calls these “rounds”. In a basic execution with only one target (and only one target port in TCP/UDP modes), the number of rounds matches the number of probes sent to the target host. However, in more complex executions where Nping is run against multiple targets and multiple ports, the number of rounds is the number of times that Nping sends a complete set of probes that covers all target IPs and all target ports. For example, if Nping is asked to send TCP SYN packets to hosts 192.168.1.0–255 and ports 80 and 433, then $256 \times 2 = 512$ packets are sent in one round. So if you specify **-c 100**, Nping will loop over the different target hosts and ports 100 times, sending a total of $256 \times 2 \times 100 = 51200$ packets. By default Nping runs for 5 rounds. If a value of 0 is specified, Nping will run continuously.

-e *name*, **--interface** *name* (Set the network interface to be used) .

This option tells Nping what interface should be used to send and receive packets. Nping should be able to detect this automatically, but it will tell you if it cannot. *name* must be the name of an existing network interface with an assigned IP address.

--privileged (Assume that the user is fully privileged) .

Tells Nping to simply assume that it is privileged enough to perform raw socket sends, packet sniffing, and similar operations that usually require special privileges. By default Nping quits if such operations are requested by a user that has no root or administrator privileges. This option may be useful on Linux, BSD or similar systems that can be configured to allow unprivileged users to perform raw-packet transmissions. The **NPING_PRIVILEGED** environment variable may be set as an alternative to using **--privileged**.

--unprivileged (Assume that the user lacks raw socket privileges) .

This option is the opposite of **--privileged**. It tells Nping to treat the user as lacking network raw socket and sniffing privileges. This is useful for testing, debugging, or when the raw network functionality of your operating system is somehow broken. The **NPING_UNPRIVILEGED** environment variable may be set as an alternative to using **--unprivileged**.

--send-eth (Use raw ethernet sending) .

Asks Nping to send packets at the raw ethernet (data link) layer rather than the higher IP (network) layer. By default, Nping chooses the one which is generally best for the platform it is running on. Raw sockets (IP layer) are generally most efficient for Unix machines, while ethernet frames are required for Windows operation since Microsoft disabled raw socket support. Nping still uses raw IP packets despite this option when there is no other choice (such as non-ethernet connections).

--send-ip (Send at raw IP level) .

Asks Nping to send packets via raw IP sockets rather than sending lower level ethernet frames. It is the complement to the **--send-eth** option.

--bpf-filter filter spec --filter filter spec (Set custom BPF filter) .

This option lets you use a custom BPF filter. By default Nping chooses a filter that is intended to capture most common responses to the particular probes that are sent. For example, when sending TCP packets, the filter is set to capture packets whose destination port matches the probe's source port or ICMP error messages that may be generated by the target or any intermediate device as a result of the probe. If for some reason you expect strange packets in response to sent probes or you just want to sniff a particular kind of traffic, you can specify a custom filter using the BPF syntax used by tools like tcpdump.. See the documentation at <http://www.tcpdump.org/> for more information.

-H, --hide-sent (Do not display sent packets) .

This option tells Nping not to print information about sent packets. This can be useful when using very short inter-probe delays (i.e., when flooding), because printing information to the standard output has a computational cost and disabling it can probably speed things up a bit. Also, it may be useful when using Nping to detect active hosts or open ports (e.g. sending probes to all TCP ports in a /24 subnet). In that case, users may not want to see thousands of sent probes but just the replies generated by active hosts.

-N, --no-capture (Do not attempt to capture replies) .

This option tells Nping to skip packet capture. This means that packets in response to sent probes will not be processed or displayed. This can be useful when doing flooding and network stack stress tests. Note that when this option is specified, most of the statistics shown at the end of the execution will be useless. This option does not work with TCP Connect mode.

OUTPUT OPTIONS**-v[level], --verbose [level]** (Increase or set verbosity level) .

Increases the verbosity level, causing Nping to print more information during its execution. There are 9 levels of verbosity (-4 to 4). Every instance of **-v** increments the verbosity level by one (from its default value, level 0). Every instance of option **-q** decrements the verbosity level by one.

Alternatively you can specify the level directly, as in **-v3** or **-v-1**. These are the available levels:

Level â4

No output at all. In some circumstances you may not want Nping to produce any output (like when one of your work mates is watching over your shoulder). In that case level â4 can be useful because although you won't see any response packets, probes will still be sent.

Level â3

Like level â4 but displays fatal error messages so you can actually see if Nping is running or it failed due to some error.

Level â2

Like level â3 but also displays warnings and recoverable errors.

Level â1

Displays traditional run-time information (version, start time, statistics, etc.) but does not display sent or received packets.

Level 0

This is the default verbosity level. It behaves like level â1 but also displays sent and received packets and some other important information.

Level 1

Like level 0 but it displays detailed information about timing, flags, protocol details, etc.

Level 2

Like level 1 but displays very detailed information about sent and received packets and other interesting information.

Level 3

Like level 2 but also displays the raw hexadecimal dump of sent and received packets.

Level 4 and higher

Same as level 3.

-q[level], --reduce-verbosity [level] (Decrease verbosity level) .

Decreases the verbosity level, causing Nping to print less information during its execution.

-d[level] (Increase or set debugging level) .

When even verbose mode doesn't provide sufficient data for you, debugging is available to flood you with much more! As with the **-v**, debugging is enabled with a command-line flag **-d** and the debug level can be increased by specifying it multiple times. There are 7 debugging levels (0 to 6). Every instance of **-d** increments debugging level by one. Provide an argument to **-d** to set the level directly; for example **-d4**.

Debugging output is useful when you suspect a bug in Nping, or if you are simply confused as to what Nping is doing and why. As this feature is mostly intended for developers, debug lines aren't always self-explanatory. You may get something like

```
NSOCK (1.0000s) Callback: TIMER SUCCESS for EID 12; tcpconnect_event_handler(): Received callback of type
```

If you don't understand a line, your only recourses are to ignore it, look it up in the source code, or request help from the development list (nmap-dev). Some lines are self-explanatory, but the messages become more obscure as the debug level is increased. These are the available levels:

Level 0

Level 0. No debug information at all. This is the default level.

Level 1

In this level, only very important or high-level debug information will be printed.

Level 2

Like level 1 but also displays important or medium-level debug information

Level 3

Like level 2 but also displays regular and low-level debug information.

Level 4

Like level 3 but also displays messages only a real Nping freak would want to see.

Level 5

Like level 4 but it enables basic debug information related to external libraries like Nsock..

Level 6

Like level 5 but it enables full, very detailed, debug information related to external libraries like Nsock.

BUGS

Like its author, Nping isn't perfect. But you can help make it better by sending bug reports or even writing patches. If Nping doesn't behave the way you expect, first upgrade to the latest Nmap version available from <http://nmap.org/download.html>. If the problem persists, do some research to determine whether it has already been discovered and addressed. Try searching for the error message on our search page at <http://insecure.org/search.html> or at Google. Also try browsing the nmap-dev archives at <http://seclists.org/>. Read this full manual page as well. If nothing comes out of this, mail a bug report to

nmap-dev@insecure.org. Please include everything you have learned about the problem, as well as what version of Nping you are running and what operating system version it is running on. Problem reports and Nping usage questions sent to nmap-dev@insecure.org are far more likely to be answered than those sent to Fyodor directly. If you subscribe to the nmap-dev list before posting, your message will bypass moderation and get through more quickly. Subscribe at <http://cgi.insecure.org/mailman/listinfo/nmap-dev>.

Code patches to fix bugs are even better than bug reports. Basic instructions for creating patch files with your changes are available at <https://svn.nmap.org/nmap/HACKING>. Patches may be sent to nmap-dev (recommended) or to any of the authors listed in the next section directly.

AUTHORS

Luis MartinGarcia luis.mgarc@gmail.com (<http://aldabaknocking.com>)

Fyodor fyodor@insecure.org (<http://insecure.org>)

NOTES

1. official type numbers assigned by IANA
<http://www.iana.org/assignments/icmp-parameters>
2. official numbers assigned by IANA
<http://www.iana.org/assignments/arp-parameters/>
3. official numbers listed by the IEEE
<http://standards.ieee.org/regauth/ethertype/eth.txt>

NAME

omniutil – collection of HP DataProtector enhancement utilities

SYNOPSIS

edrc/lib/omniutil [**-h**]

omniutil **-a pd_create** **-i ora_sid** **-d basedir** [**-v ora_vers**]

omniutil **-a pd_link** **-i ora_sid** **-d basedir**

AVAILABILITY

WA2L/edrc

DESCRIPTION

omniutil is a collection of HP DataProtector enhancement tools.

The different tools can be started using the **-a action** option.

The remaining options are dependent of the selected *action*.

For a description of the functionality of the different *actions* see section **OPTIONS**.

OPTIONS

-h usage message.

-a action :

pd_create

create a pointer directory structure. This is a structure mainly consisting of symbolic links that will point to an **\$ORACLE_HOME** after invoking **-a pd_link** later.

This enables DataProtector to backup databases, that have no own **\$ORACLE_HOME** installation. The **\$ORACLE_HOME** setting in DataProtector can be set to this "pretended" **\$ORACLE_HOME**.

To allow a pseudo-dynamic setup, after invoking **-a pd_create** ... the two symbolic links (**.ora_home** and **.network_admin**) in the applied structure have to be set accordingly using **-a pd_link**

See also section **EXAMPLES**.

pd_link

the structure applied using **-a pd_create** ... mainly contains symbolic links that first point to the two main symbolic links (**.ora_home** and **.network_admin**). This two links then should point to the real **\$ORACLE_HOME**. This indirect symbolic link structure enables an automated update of the structure if an Oracle installation is patched creating a new **\$ORACLE_HOME** with the applied patches. Normally the directory structure after installing a patch is identical, but the binaries etc. might change. If a patch is applied using this method it is only needed to update the two symbolic links **.ora_home** and **.network_admin** using the **-a pd_link** ... option.

In the configuration file **omniutil.cfg** the command to resolve the current **\$ORACLE_HOME** and **\$TNS_ADMIN** can be specified in the **ORACLE_ENV_CMD** setting. If it is not possible to resolve this environment variables, they can be set manually previous to the **omniutil** command invocation using **export ORACLE_HOME=path** and **export TNS_ADMIN=path**.

To ensure that the described symbolic links always point to the most recent **\$ORACLE_HOME** the **~edrc/bin/omniutil -a pd_link -i ora_sid -d basedir** can be defined as **PRE_EXEC** script in the DataProtector Barlist for the related database backup.

See also section **EXAMPLES**.

-i ora_sid Oracle SID.

-d basedir Base directory of the pointer directory structure.

-v ora_version

Database version (=template file **ora_version.tar.gz** located in **edrc/var/omniutil/pd/**).

ENVIRONMENT

-

EXIT STATUS

- | | |
|-----------|---|
| 0 | no error. |
| 2 | operating system not supported, see osid(3) if you get this error. |
| 4 | usage listed. |
| 6 | configuration file omniutil.cfg does not exist. |
| 10 | template for the specified Oracle version (ora_vers) does not exist. |
| 11 | DataProtector Oracle pointer directory (<basedir>/<ora_sid>) already exists. |

- 12 template does not contain the directory for the specified Oracle version (*ora_vers*).
- 13 cannot write to the specified *basedir*.
- 21 DataProtector pointer directory **<basedir>/<ora_sid>** does not exist.
- 23 cannot write to the specified *basedir*.

FILES

edrc/etc/omniutil.cfg

configuration file for **omniutil**.

edrc/var/omniutil/pd/<ora_vers>/

template files for the pointer directory structure.

edrc/var/omniutil/pd/<ora_vers>.tar.gz

packed template files for the pointer directory structure. For easy distribution of the template structure, always the packed version of the structure template should be used.

<basedir>/<ora_sid>/

applied template pointer directory structure

<basedir>/<ora_sid>/ora_home

main symbolic link of applied template pointer directory structure pointing to the **\$ORACLE_HOME** directory of the installed Oracle software for a certain database (*ora_sid*).

<basedir>/<ora_sid>/network_admin

main symbolic link of applied template pointer directory structure pointing to the **\$TNS_ADMIN** directory of the installed Oracle software for a certain database (*ora_sid*).

EXAMPLES

1) create a pointer directory structure

This example shows how to create a pointer directory structure for the database *FS90DMO* in the directory **/data_ps1/dat/omni/** that will be used as **\$ORACLE_HOME** for this database in DataProtector. The **\$ORACLE_HOME** that is used by the database itself is located in **/ora/product/10204**.

1.1) create the directory structure

Create the generic pointer directory structure for the database version *10gR2* :

```
[ /data_ps1/dat/omni ]
[ root@acme001 ] [-sh]: ~edrc/bin/omniutil -a pd_create -i FS90DMO \
                        -d /data_ps1/dat/omni -v 10gR2
```

omniutil - DataProtector Utilities

```
create OBII Oracle pointer directory structure ...
  create /data_ps1/dat/omni/FS90DMO ... done.
  create structure ...
    using template: /opt/edrc/var/omniutil/pd/10gR2.tar.gz
    /data_ps1/dat/omni/FS90DMO/.patch_storage
    /data_ps1/dat/omni/FS90DMO/OPatch
    /data_ps1/dat/omni/FS90DMO/assistants
    /data_ps1/dat/omni/FS90DMO/bin
    /data_ps1/dat/omni/FS90DMO/cdata
    /data_ps1/dat/omni/FS90DMO/cfgtoollogs
    /data_ps1/dat/omni/FS90DMO/clone
    /data_ps1/dat/omni/FS90DMO/config
    /data_ps1/dat/omni/FS90DMO/crs
    /data_ps1/dat/omni/FS90DMO/css
    /data_ps1/dat/omni/FS90DMO/ctx
    /data_ps1/dat/omni/FS90DMO/dbs
    /data_ps1/dat/omni/FS90DMO/demo
    /data_ps1/dat/omni/FS90DMO/diagnostics
    /data_ps1/dat/omni/FS90DMO/has
    /data_ps1/dat/omni/FS90DMO/hs
    /data_ps1/dat/omni/FS90DMO/install
    /data_ps1/dat/omni/FS90DMO/inventory
  :
  :
    /data_ps1/dat/omni/FS90DMO/sqlj
    /data_ps1/dat/omni/FS90DMO/sqlplus
    /data_ps1/dat/omni/FS90DMO/srvn
    /data_ps1/dat/omni/FS90DMO/sysman
    /data_ps1/dat/omni/FS90DMO/tg4ifmx
    /data_ps1/dat/omni/FS90DMO/tg4ingr
    /data_ps1/dat/omni/FS90DMO/tg4sybs
    /data_ps1/dat/omni/FS90DMO/tg4tera
    /data_ps1/dat/omni/FS90DMO/uix
    /data_ps1/dat/omni/FS90DMO/wwg
    /data_ps1/dat/omni/FS90DMO/xdk
    /data_ps1/dat/omni/FS90DMO/.network_admin
    /data_ps1/dat/omni/FS90DMO/.ora_home
    0 blocks
  done.
done.
```

1.2) set the directory structure link

Update the symbolic links to point to the **\$ORACLE_HOME** for the given database:

```
[ /data_ps1/dat/omni/FS90DMO ]
[ root@acme001 ][-sh]: ~edrc/bin/omniutil -a pd_link -i FS90DMO \
                        -d /data_ps1/dat/omni
```

omniutil - DataProtector Utilities

```
update OBII Oracle pointer directory symlinks ...
```

```
in directory /data_psl/dat/omni/FS90DMO ...
omniutil-INFO: symlink '.network_admin -> /ora/product/10204/netw
omniutil-INFO: symlink '.ora_home -> /ora/product/10204' created
done
done.
```

SEE ALSO

catcomp(1), **edrcintro(1)**, **gzip(1)**, **lscomp(1)**, **llcomp(1)**, **omniutil.cfg(4)**, **tar(1)**

NOTES

-

BUGS

-

AUTHOR

omniutil was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

omniutil.cfg – configuration file for omniutil

SYNOPSIS

edrc/etc/omniutil.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **omniutil** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **omniutil**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/omni/log

Default: LOG='approot'/var/log

OBII_DEFAULT_ORACLE_VERSION

Default Oracle version if no version is specified in the command line. This setting specifies the template used in **edrc/var/omniutil/pd**.

Example: OBII_DEFAULT_ORACLE_VERSION=Oracle9i

Default: OBII_DEFAULT_ORACLE_VERSION=Oracle8

ORACLE_ENV_CMD

Command used to evaluate the current **\$TNS_ADMIN** and **\$ORACLE_HOME** of a database. The variables **\$OBII_ORACLE_VERSION** and **\$OBII_ORACLE_INSTANCE** can be used as options in the **ORACLE_ENV_CMD**.

Example: ORACLE_ENV_CMD='./usr/local/bin/sid \$OBII_ORACLE_INSTANCE'

Default: ORACLE_ENV_CMD='ini.ksh \$OBII_ORACLE_VERSION'

EXEC_SHELL

Shell to execute the **ORACLE_ENV_CMD**.

Example: EXEC_SHELL=/bin/ksh

Default: EXEC_SHELL=/bin/ksh

SEE ALSO

omniutil(1), **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

omniutil.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`os_wrapper` – wrap operating system dependent binaries

SYNOPSIS

`edrc/bin/command -> .os_wrapper`

`edrc/lib/command -> .os_wrapper`

AVAILABILITY

WA2L/edrc

DESCRIPTION

wrap operating system dependent binaries to allow transparent command startup.

If a binary for the related operating system (OSID) is installed in WA2L/edrc this binary is started. When the binary is not installed in WA2L/edrc, but the operating system provides it, it is started from the operating system.

This default *command* startup behaviour can be altered in the configuration file **lib/os_wrapper/command.cfg** for each *command* started thru the **.os_wrapper**.

The concept behind the default startup behaviour is, that on certain operating systems some commands are not installed by default. Therefore only for those operating systems the "missing" commands have to be distributed with WA2L/edrc, in this case the command provided in WA2L/edrc complements the commands provided by the operating system.

Furthermore it might also be the case that some variants of commands provided by a certain operating systems do not provide the needed functionality, in this case the command variant provided in WA2L/edrc over-locks the one installed by the operating system.

An example for over-locking is the **cpio** command for HP-UX, where the operating system command lacks some options and is not able to copy files bigger then 2 GByte.

An example for complementing is the **traceroute** command for SunOS, where it was not installed by default on early Solaris releases.

When a binary that is provided in WA2L/edrc is started thru the **.os_wrapper** the **\$LD_LIBRARY_PATH** and based on the underlying operating system, the **\$SHLIB_PATH** environment variables are expanded with **lib/<OSID>/libs/**.

To start a new command thru the **.os_wrapper**, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ] [*edrc*/bash]: ln -s .os_wrapper new_cmd
```

- 2.) install the command binary in the **edrc/bin/<OSID>/** or **edrc/lib/<OSID>/** directory depending on your needs and set the correct file ownership (0:3) and permissions (*executable*):

```
[ /opt/edrc/lib/Linux ]
[ root@acme001 ][*edrc*/bash]: chown root:3 new_cmd

[ /opt/edrc/lib/Linux ]
[ root@acme001 ][*edrc*/bash]: chmod 755 new_cmd
```

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- 103** user calling the command has no permission to access/execute the called *daemon_command*.
- 106** the operating system id (OSID) is not supported, yet. If you get this error, the *command* is neither installed in WA2L/edrc, nor on the underlying operating system.
- 107** the **.os_wrapper** was called directly.
- other** exit status of the started *command*.

FILES

- lib/os_wrapper/command.cfg**
configuration file for the *command* started thru the **.os_wrapper**. If this file is missing for a *command*, the default startup behaviour applies.
- bin/<OSID>/command**
location of the provided *command* for a certain operating system id (OSID).
- bin/<OSID>/<GLIBC>/command**
location of the provided *command* for a certain operating system id (OSID) and for a specific GLIBC version.

The GLIBC version can be resolved using the **glibc.version(3)**, the **ldd --version** or the **get-conf GNU_LIBC_VERSION** command.
- lib/<OSID>/command**
location of the provided *command* for a certain operating system id (OSID).

lib/<OSID>/<GLIBC>/command

location of the provided *command* for a certain operating system id (OSID) and for a specific GLIBC version.

The GLIBC version can be resolved using the **glibc.version(3)** or the **ldd --version** command.

lib/<OSID>/libs/

location of the provided shared libraries for a certain operating system id (OSID).

lib/<OSID>/<GLIBC>/libs/

location of the provided shared libraries for a certain operating system id (OSID) and for a specific GLIBC version.

EXAMPLES

-

SEE ALSO

edrcintro(1), **binprobe(1m)**, **cmdlist(1m)**, **daemon_wrapper(1)**, **glibc.version(3)**, **java_wrapper(1)**, **ksh_wrapper(1)**, **lua_wrapper(1)**, **ln(1)**, **ld(1)**, **ldd(1)**, **osid(3)**, **perl_wrapper(1)**, **shlib(3)**

NOTES

-

BUGS

-

AUTHOR

os_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

osid – evaluate the system's operating system id

SYNOPSIS

edrc/lib/osid [**-h** | **-p** | **-s** | **-e**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

osid generates an unique identifier for each combination of OS release (**uname -r**) and OS version (**uname -s**). This identifier is used, particularly by **sysconfig**, to identify the appropriate files for system configuration. Furthermore many scripts use **osid** to make it compatible to multiple operating systems.

This tool is necessary because the OS alone is not sufficient to uniquely identify the proper configuration files for an OS. For example, both Solaris 1.x and 2.x call themselves "SunOS". But because Solaris 1.x is a BSD release and Solaris 2.x is a System V release, they require completely different configuration files.

To minimize the risk of damaging a future, incompatible release of an existing operating system, **osid** will only return a value different from *unknown* if the entry in **osid.dat** exactly matches the values returned by **uname -rs**.

If two releases of an Operating System are sufficiently similar (for instance HP-UX v9.01 and v9.03), they may of course be mapped to the same OSID in **osid.dat**.

If the file **/etc/sfi/OSID** exists, the value from this file is returned by **osid**. If this file does not exist, the current OSID is resolved from the **edrc/etc/osid.dat** file.

OPTIONS

- h** print **osid** usage.
- p** print the OSID definitions as an unordered list identical to the sequence in the **osid.dat** file. This output can also be used to debug the definitions in an **osid.dat** file due to the fact that OSID of the first matching entry will be returned as the resolved OSID.
- s** print the OSID definitions as an ordered list.
- e** print a line in the **osid.dat** format based on the current operating system. This option helps to quickly add a line to the related **osid.dat** file. The fields *unknown* and *<Comment>* have to be replaced by the correct values.

EXIT STATUS

- 0** the operating system id could be resolved.

- 1 the operating system id could not be resolved. In this case **osid** returns *unknown* as OS-id. If you receive this return value you should check the configuration file **edrc/etc/osid.dat** if the entry for the current operating system is missing. If two releases on an operating system are sufficient similar, they should be mapped to the same OS-id.
- 2 the option **-p** or **-s** has been invoked.
- 4 usage listed.
- 5 the option **-e** has been invoked.

EXAMPLES

- 1) HOWTO use osid to write a multiple os compatible script

```

:
:

case "`osid`" in
    Solaris)
        DF="df -k"
        AWK=nawk
        ECHO=echo
        ;;
    HP-10|HP-11)
        DF="bdf"
        AWK=awk
        ECHO=echo
        ;;
    Linux)
        DF="df -k"
        AWK=awk
        ECHO="echo -e"
        ;;
    unknown)
        echo "example-FATAL: OS not defined"
        exit 2
        ;;
    *)
        echo "example-FATAL: OS not supported"
        exit 2
        ;;
esac

:
:

cd /
$DF | $AWK '{ printf("%10d | %10d | %s\n", $3, $4, $6)}'

:
:
```

SEE ALSO

edrcintro(1), **mandir**(4), **osid.cfg**(4), **osid.dat**(4), **osid.probe**(3), **rosid**(3), **sysconfig**(1)

FILES

/etc/sfi/OSID

operating system id on a system configured by **rmtconfig** of the **SFI-Director**.

edrc/etc/osid.cfg

optional configuration file for **osid**. See **osid.cfg**(4) for more information.

edrc/etc/osid.dat

mapfile of **uname -rs** or **uname -rsm** to OS-id. See **osid.dat**(4) for more information.

edrc/lib/OSID/osid.probe

the **osid.probe** command is a C-program that is compiled for each OSID. It is used to probe if the binary related to the resolved OSID is able to run on the system. If the **osid.probe** binary can't be executed it is assumed that also the other binaries related to the OSID aren't compatible to the system architecture and *unknown* is returned by the **osid** command
If the **osid.probe** command does not exist the binary compatibility is not checked.

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

Due to the fact that certain operating system vendors started to distribute identical operating system releases (**uname -r**) on different platforms, the need came up to even distinguish within identical operating system releases between the platforms. Therefore the machine (hardware) type (**uname -m**) can be considered too. This lead into an extended format of the **osid.dat** file, that usage can be configured in the **osid.cfg** configuration file, if needed.

BUGS

-

AUTHOR

osid was initially developed by Peter Stevens. The version included in **WA2L/edrc** has been rewritten by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2011 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

osid.cfg – configuration file for osid

SYNOPSIS

edrc/etc/osid.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the optional configuration file for the **osid** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the **=** and the **VALUE** are no spaces.

You should not comment out any **OPTION**. If you like to use default settings simply do not specify a **VALUE**.

OPTIONS**OSID_DAT_FORMAT**

Format of the **osid.dat** file.

The **SFI** format as known from the SFI Director in the years 1996-2008 is:
OS:release:OSID:Comment

The extended **WA2L** format starting from 2008 is: *OS:release:machine:OSID:Comment*

See **osid.dat(4)** for more information.

Default is the **SFI** format. The **SFI** format also applies when the **osid.cfg** file does not exist.

Example: **OSID_DAT_FORMAT=WA2L**

Default: **OSID_DAT_FORMAT=SFI**

OSID_DAT_FILE

Name of the **osid.dat** file in the **edrc/etc** directory.

The default if the specified file does not exist or if the **osid.cfg** file does not exist is **osid.dat**.

Example: OSID_DAT_FILE=osid.dat.wa2l

Default: OSID_DAT_FILE=osid.dat

PROBE_BINARY

With this setting the probing of binaries for the resolved OSID can be turned off. Currently I cannot imagine a reason why to set this option to False, but you never know.

Example: PROBE_BINARY=False

Default: PROBE_BINARY=True

DISABLE_REGEX

If the **OSID_DAT_FORMAT** is set to **WA2L**, the first three fields in the specified **OSID_DAT_FILE** (<OS>:<release>:<machine>:) are treated as regular expressions. To disable this feature, set **DISABLE_REGEX** to **True**.

Example: DISABLE_REGEX=True

Default: DISABLE_REGEX=False

SEE ALSO

edrcintro(1), **osid**(3), **osid.dat**(4)

NOTES

-

BUGS

-

AUTHOR

osid.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

osid.dat – operating system versions map for osid

SYNOPSIS

edrc/etc/osid.dat
edrc/etc/osid.dat.WA2L

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the OSID map file for the **osid** command.

There are two formats of the **osid.dat** file possible. One (the file format known from the SFI-Director, that suited the requirements for more then ten years from 1996 to 2008) considers the **uname -rs** combinations. The new slightly extended format (needed due to the operating system vendors started to distribute identical operating system versions on different platforms) considers the **uname -rsm** combinations.

The used format and the **osid.dat** file to be used for your environment can be configured in the **osid.cfg(4)** configuration file.

The default is still the **SFI** format.

SFI FORMAT

This list defines a unique identifier for each combination of OS release (**uname -r**) and OS version (**uname -s**). It maps these two values to OSID. This is defined to separate different OS and release levels. If two different releases are sufficiently close to each other, they may be mapped into the same OSID. The command **osid** processes a **uname -rs** and compares the result with the contents of **edrc/etc/osid.dat**. If the result matches to an entry, the corresponding OSID is returned.

WA2L FORMAT

This list defines a unique identifier for each combination of OS release (**uname -r**), OS version (**uname -s**) and machine type (**uname -m**). It maps these three values to OSID. This is defined to separate different OS, release levels and platform types. If two different releases are sufficiently close to each other, they may be mapped into the same OSID. The command **osid** processes a **uname -rsm** and compares the result with the contents of **edrc/etc/osid.dat**. If the result matches to an entry, the corresponding OSID is returned.

Due to the fact, that especially on Linux operating systems, the **uname -r** output changes also when patches are installed and not only on operating system release upgrades as on Solaris or HP-UX, the first three fields of the **edrc/etc/osid.dat** file in the **WA2L** format are treated as regular expressions, by default. This enables, to resolve the OSID more reliable.

Be aware, that all regular expressions in the file are correct, else the OSID resolution might fail.

Definition example using regular expressions:

```
Linux:2.6.35.[0-9][0-9]-[0-9][0-9].fc14.i686:i686:Linux:Fedora Core 14:
```

This definition does the job, but is not completely correct due to the fact that a dot (.) allows to be replaced by any character and therefore should be escaped using the backslash (\). However, the danger to map a wrong OSID to the operating system not escaping the dots might often be low and you have to decide by your own, if it is more important to have a more easy to read definition, not escaping the dots, or to have a more correct definition, by escaping the dots. For more info to regular expressions **regexintro(4)**.

Definition example using regular expressions strictly:

```
Linux:2\.6\.35\.[0-9][0-9]-[0-9][0-9]\.fc14\.i686:i686:Linux:Fedora Core 14
```

Definition example not using regular expressions:

```
Linux:2.6.35.11-89.fc14.i686:i686:Linux:Fedora Core 14:
Linux:2.6.35.12-90.fc14.i686:i686:Linux:Fedora Core 14:
Linux:2.6.35.13-91.fc14.i686:i686:Linux:Fedora Core 14:
```

FILEFORMAT

Rows starting with a # are considered as comments.

SFI FORMAT

OS:release:OSID:Comment

WA2L FORMAT

OS:release:machine:OSID:Comment

KEYS

| | |
|----------------|---|
| <i>OS</i> | Name of the operating system, output of uname -s . |
| <i>release</i> | Operating system release level, output of uname -r . |
| <i>machine</i> | machine (hardware) type, output of uname -m . |
| <i>OSID</i> | Operating system identifier, defined by the WA2L/edrc package or the SFI Director. There might be a file named /etc/sfi/OSID on each host maintained by the SFI Director, containing the OSID from ~sfiadmin/osid.dat . |
| <i>Comment</i> | An optional comment. This comment is currently not resolved and is only used for human reading. |

SEE ALSO

edrcintro(1), **osid(3)**, **osid.cfg(4)**, **regexintro(4)**

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

BUGS

-

AUTHOR

osid.dat was developed by Peter Stevens and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

osid.probe – probe if chosen OSID matches to binary format

SYNOPSIS

edrc/lib/osid.probe
edrc/lib/<OSID>/osid.probe

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **osid.probe** is a simple compiled C program that is called by the **osid** command to check, if the chosen OSID matches to the binary format. If the **osid.probe** command can be executed, it's intended that the resolved OSID is correct.

osid.probe returns True, if the **OSID_PROBE** environment variable is not set.

OPTIONS

-

ENVIRONMENT**\$OSID_PROBE**

if the environment variable is set to **False**, the text and return-code as if the binary format is not compatible to the operating system is returned. This environment variable is intended to simulate incompatibility.

EXIT STATUS

0 no error.

1 cannot execute binary file. The chosen OSID is not correct, due to the fact that the probe command **osid.probe** cannot be executed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **osid**(3), **osid.cfg**(4), **osid.dat**(4)

NOTES

-

BUGS

-

AUTHOR

osid.probe was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2011 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

osup – start edrc with a special Operation Support configuration

SYNOPSIS

edrc/bin/osup [**-h** | **-V**]

osup [**-s**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Short start of **edrc** with an other configuration which points to an own script tree. Internally **osup** calls **edrc -c edrc.osup.cfg -n OSUP_@ID@** .

osup stands for "Operation SUPport".

The session name (as shown in the menu) is automatically set to **OSUP_<id>** . Where **<id>** is the process id of the started **edrc** instance if not already a session with the same name exists, if so the **<id>** is set to a random number.

If additional short starts are needed, create a symlink from the new short start command to **sat** (see section **EXAMPLES**).

OPTIONS

-h usage message.

-V print version and patch level of **edrc**. For an explanation of the release numbering system see **edrcrevision(1)**.

-s silent startup. Startup without showing the EDRC banner.

-t no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

ENVIRONMENT

-

EXIT STATUS

see **edrc**(1m).

FILES

etc/edrc.osup.cfg

configuration file of **osup**, see **edrc.cfg**(4) for more information.

Other files see section **FILES** in **edrc**(1m).

EXAMPLES**1) create a new short start**

This creates a new short start command **new_shortstart** that will load the configuration file **edrc.new_shortstart.cfg**. The session name will automatically be set to **NEW_SHORTSTART_<id>** .

```
[ /root ]
[ root@rh7mzv7t001 ] [bash]: cd ~edrc/bin

[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ln -s sat new_shortstart
```

SEE ALSO

sat(1), **edrc**(1m), **edrc.cfg**(4), **edrcintro**(1), **edrcrevision**(1)

NOTES

-

BUGS

-

AUTHOR

osup was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

outex – produce contrib.doc OUTPUT-EXAMPLE output from logfile

SYNOPSIS

edrc/bin/outex [**-h**]

outex [*logfile*]

outex last

AVAILABILITY

WA2L/edrc

DESCRIPTION

produce output of a selected logfile (without path) that is saved in **edrc/var/log/** that can be used in **contrib.doc**(1m) in the **OUTPUT-EXAMPLE** section.

The intension is to use the **outex** *logfile* command directly from within **vi** when editing a recovery script, as:

```
# D: Restore database.
#
~
~
~
: . !outex 2020-06-02_10.52.05__db_restore.log
```

This will include the contents of the logfile **edrc/var/log/2020-06-02_10.52.05__db_restore.log** (without logfile header) into the **vi** session. Each line will be prepended by the **# O:** documentation tag.

The specified *logfile* can also be a gzipped file, as: **2020-06-02_10.52.05__db_restore.log.gz**

Furthermore if the specified logfile **2020-06-02_10.52.05__db_restore.log** does not exist, but the **2020-06-02_10.52.05__db_restore.log.gz** does, the contents of the compressed file is included and vice versa.

After receiving the log data in **vi**, press **[Ctrl] + [L]** to redraw the terminal.

OPTIONS

-h usage message.

 specify the *logfile* interactively.

logfile logfile to be read.

The file path must not be specified, due to the fact that **outex** only computes log files from the **edrc/var/log/** directory; if a path is specified, the path part of the input is ignored.

The *logfile* can be a gnu-zipped log file (**file.log.gz**) as supported by **edrc**, it is not needed to decompress the file. Furthermore if the specified *file.log* does not exist, but the *file.log.gz* does, the *.**gz** file is loaded and vice versa.

last load the last (newest) available recovery script **logfile** from **edrc/var/log**.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|--------------------------------|
| 4 | usage printed. |
| 1 | <i>logfile</i> not found. |
| 2 | <i>logfile</i> cannot be read. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **contrib.doc(1m)**, **contrib.edrc(1m)**, **tpl(1)**

NOTES

-

BUGS

-

AUTHOR

outex was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pack – pack an application to a software package

SYNOPSIS

edrc/bin/pack [**-h**]

pack [**-a** *app*] [**-f**] [**-s**] [**-d** *outdir*] [**-o** *package_format*] [**-c** *config_file*] [**-t**]

pack [**-a** *app*] [**-f**] [**-s**] [**-d** *outdir*] [**-o** *package_format*] [**-c** *config_file*] **-p** *version*

pack -L [**-a** *app*] [**-c** *config_file*] [**-p** *version*]

pack -l [**-c** *config_file*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Create a software package of an installed application.

A software package should not contain logfiles, backup files, lock files etc. To support this, it can be defined in the configfile of **pack** which files should be excluded from the software package.

pack enables the user to pack an application that has been installed (and probably configured) back to an installable package without the need of manually copy and remove certain files. Furthermore the user does not need to know all fine details of the application e.g. which files are needed for a correct operation, which files can be deleted, which files would harm a correct operation on an other system if packed.

It is possible to create a full RELEASE or a PATCH of the current state to a certain version.

A RELEASE is created if the **-p version** option is not specified and contains all files of the application.

A PATCH is created if the **-p version** option is specified and contains all files that are newer then the point in time when the RELEASE has been created for a specific *version* of the application.

In this context it is important to realize that if files are added to an application which are older then the timestamp of the last RELEASE, those files are not added to the PATCH. This is often the case if **.tar.gz** files are downloaded and the untared files are added to the application due to the fact that the file timestamps are preserved. To ensure that those files are added to the PATCH a **touch filename** should be issued for all files added to the application. It might also be advisable to check the filepermissions and the ownership of the files.

OPTIONS

Despite of the many options available, it is often not needed to provide any option to **pack** due to the fact that a developer should define in the **default** application setting the common case for the application package creation. Issue **pack -l** to list all available definitions.

- h** usage message.
- a** *application* application handle. If this option is not specified *default* is used as the application handle.
- s** split package also in pieces of 1400000 bytes which fit to a diskette. This option can also be used to mail the package when mail size restrictions apply on sender or receiver side.

If this option is selected beside the pieces a setup script is provided to concatenate the pieces together in the correct order and to check if all pieces are available. Furthermore some instructions to the user are printed.
- f** force overwriting of output file if a file with the identical name already exists in the output directory.
- p** *version* patch application to the specified version. The timestamp for the version specified has to exist. A timestamp is always written if a **RELEASE** is generated and the **-t** option is specified. To install a patch, use the **patchinstall(3)** command.
- l** list all applications defined in the configuration file. The available timestamps to create patches are listed, too.
- L** list files to be packed into the software package without actually creating the package file. This option is useful to verify if your settings of **EXCLUDE_LIST** in the configuration file are correct.
- d** *output_dir* directory where to put the package file. If not specified this defaults to the setting made in **OUTPUT_DIRECTORY** of the related application specified with the **-a** *application* option.
- o** *package_format* format of the software package to be created. If not specified the software package format defaults to **shar**.
 - shar** generic installable shell archive. A shell archive is a file that contains the setup procedure and the software data. This can be compared with a **setup.exe** program often used on the Microsoft Windows (TM) operating system. See section **EXAMPLES** or issue **pack -h** for an example usage of a shell archive.
 - depot** HP-UX software depot. This format is used on Hewlett Packard HP-UX systems. The installation of a **depot** package is done with the **swinstall** command on HP-UX. The generation of this package format is currently not implemented.
 - pkg** Solaris package. This format is used on Sun Solaris systems. The installation of a **pkg** package is done with the **pkgadd** command on Solaris. The generation of this package format is currently not implemented.
 - rpm** RedHat package format. This format is mainly used on Linux systems. The installation of a **rpm** package is done with the **rpm** command. The generation of this package format is currently not implemented.

- c** *config_file*
configuration file for **pack**. If this option is not specified, the configuration file **edrc/etc/pack.cfg** is used.
- t**
the timestamp is updated when a RELEASE is generated (= if **pack** is called without the **-p version** option). It makes sense to use this option to produce productive RELEASES. In development cases this options won't be used.

ENVIRONMENT

\$EDRC_DEBUG_PACK_EXCLUDE

when set to **True** all regular expressions defined in the **EXCLUDE_LIST** of the **pack** configuration file and the matched files by the related regular expression are recorded to the **/tmp/pack.<DATETIME>.exclude** file.

This allows to identify which regular expression of the configuration file excludes what files.

EXIT STATUS

- 0** no error.
- 1** no application defined in configuration file.
- 2** operating system not supported. See **osid(3)** if you get this error.
- 3** application is not defined in configuration file.
- 4** usage displayed.
- 5** command aborted.
- 6** configuration file does not exist.
- 7** package format is not supported yet.
- 8** cannot write to logfile.
- 9** cannot write to output directory.
- 10** application cannot be patched against version specified. This is the case if no RELEASE of the specified application exists for the version specified.
- 11** a temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

- 12** a temporary output directory could not be claimed or created in the output directory.
- 99** creation of package format is not implemented yet.

FILES

edrc/etc/pack.cfg

default configuration file of **pack**.

edrc/var/pack/ts/<APPLICATION>-<VERSION>

timestamp of an APPLICATION for the related VERSION. Be aware that the timestamp is saved in GMT regardless of the local time zone setting.

edrc/var/pack/sadm/<APPLICATION>-<VERSION>-<DATETIME>.gz

software admin file for a RELEASE of an APPLICATION for the related VERSION. This file contains all information about the creation of a software package created on the DATE at the TIME.

edrc/var/pack/sadm/<APPLICATION>-<VERSION>-<DATETIME>-PATCH-<PATCHVERSION>.gz

software admin file for a PATCH of an APPLICATION for the related VERSION to a PATCHVERSION. This file contains all information about the creation of a software package created on the DATE at the TIME.

edrc/lib/<PACKAGE_COMPONENTS>/

package components used to generate a certain package format.

edrc/lib/<PACKAGE_COMPONENTS>/shar.header

header file for the shell archive software package. This header is displayed when installing the produced shell archive.

edrc/lib/<PACKAGE_COMPONENTS>/shar.setup

root setup script for a shell archive.

edrc/lib/<PACKAGE_COMPONENTS>/shar.install.tar.gz

install script structure for a shell archive.

edrc/lib/<PACKAGE_COMPONENTS>/piece.setup

setup script to concatenate a splitted archive.

edrc/lib/<PACKAGE_COMPONENTS>/rpm.specification

rpm specification template to generate a rpm archive.

EXAMPLES

0) create a package of the default application

In this example usage the application timestamp is not updated. This is useful if an installed application has to be packed to be installed on an other host.

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][bash]: ./pack

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...
  application information ...
    APPLICATION ..... : default
    APPLICATION_PREFIX ... : edrc
    APPLICATION_NAME ..... : WA2L/edrc
    APPLICATION_RELEASE .. : 1.5.02
    DESCRIPTION ..... : WA2L/edrc complete
  done.
  package information ...
    format ..... : shar
    type ..... : RELEASE
    file ..... : /tmp/edrc-1.5.02-200705172007.sh
  done.
  write sadm file ...(/opt/edrc/var/pack/sadm/edrc-1.5.02-200705172007.
  evaluate files to be packed ...(11699 files)... done.
  evaluate properties of files to be packed ... done.
  pack files to package file ...(28262.7 kBytes)... done.
done.
```

1) create a RELEASE of the default application

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][bash]: ./pack -t \
                                -d /dat/sw/apps/edrc-1.5.02

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...
  application information ...
    APPLICATION ..... : default
    APPLICATION_PREFIX ... : edrc
    APPLICATION_NAME ..... : WA2L/edrc
    APPLICATION_RELEASE .. : 1.5.02
    DESCRIPTION ..... : WA2L/edrc complete
  done.
  package information ...
    format ..... : shar
    type ..... : RELEASE
    file ..... : /dat/sw/apps/edrc-1.5.02/edrc-1.5.02-200
  done.
  write timestamp ...(/opt/edrc/var/pack/ts/edrc-1.5.02)... done.
  write sadm file ...(/opt/edrc/var/pack/sadm/edrc-1.5.02-200705172007.
  evaluate files to be packed ...(11699 files)... done.
  evaluate properties of files to be packed ... done.
```

```
pack files to package file ...(28262.7 kBytes)... done.
done.
```

2) create a PATCH of the current situation to version 1.5.01

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ./pack -p 1.5.01

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...
application information ...
  APPLICATION ..... : default
  APPLICATION_PREFIX ... : edrc
  APPLICATION_NAME ..... : WA2L/edrc
  APPLICATION_RELEASE .. : 1.5.02
  DESCRIPTION ..... : WA2L/edrc complete
done.
package information ...
  format ..... : shar
  type ..... : PATCH (to version 1.5.01)
  file ..... : /tmp/edrc-1.5.02-200705172015-PATCH-1.5.
done.
write sadm file ...(/opt/edrc/var/pack/sadm/edrc-1.5.02-200705172015-
evaluate files to be packed ...(3501 files)... done.
evaluate properties of files to be packed ... done.
pack files to package file ...(9038.8 kBytes)... done.
done.
```

3) list files that would be part of a PATCH of the default application to version 1.5.01

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ./pack -p 1.5.01 -L

pack - create application package, by Chr. Walther

list files to be packed of application in '/opt/edrc' ...
application information ...
  APPLICATION ..... : default
  APPLICATION_PREFIX ... : edrc
  APPLICATION_NAME ..... : WA2L/edrc
  APPLICATION_RELEASE .. : 1.5.02
  DESCRIPTION ..... : WA2L/edrc complete
done.
package information ...
  format ..... : shar
  type ..... : PATCH (to version 1.5.01)
  file ..... : NONE
done.
evaluate files to be packed ...(3500 files)... done.
files in package
  edrc/
  edrc/bin/
  edrc/bin/apply2sw_inventory
```

```

edrc/bin/asup
edrc/bin/bunzip2
:
:
edrc/var/sysconfig/tools/default/
edrc/var/sysconfig/tools/HP-11/
edrc/var/sysconfig/tools/HP-11i
edrc/var/sysconfig/tools/Linux/
edrc/var/sysconfig/tools/Solaris/
done.
done.

```

4) create a RELEASE of the default application and split also the outputfile

Here in addition to the complete shell archive **edrc-1.5.02-200705172019.sh** the splitted files **edrc-1.5.02-200705172019.sh.piece_aa ... edrc-1.5.02-200705172019.sh.piece_au** and the setup script **edrc-1.5.02-200705172019.sh.piece.sh** is generated.

```

[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ./pack -t -s \
                                -d /dat/sw/apps/edrc-1.5.02

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...
application information ...
  APPLICATION ..... : default
  APPLICATION_PREFIX ... : edrc
  APPLICATION_NAME ..... : WA2L/edrc
  APPLICATION_RELEASE .. : 1.5.02
  DESCRIPTION ..... : WA2L/edrc complete
done.
package information ...
  format ..... : shar
  type ..... : RELEASE
  file ..... : /dat/sw/apps/edrc-1.5.02/edrc-1.5.02-200
done.
write timestamp ...(/opt/edrc/var/pack/ts/edrc-1.5.02)... done.
write sadm file ...(/opt/edrc/var/pack/sadm/edrc-1.5.02-200705172019.
evaluate files to be packed ...(11699 files)... done.
pack files to package file ...(28262.7 kBytes)... done.
split 'shar' package into '1400000' pieces ...
  edrc-1.5.02-200705172019.sh.piece_aa
  edrc-1.5.02-200705172019.sh.piece_ab
  edrc-1.5.02-200705172019.sh.piece_ac
  edrc-1.5.02-200705172019.sh.piece_ad
  :
  :
  edrc-1.5.02-200705172019.sh.piece_ar
  edrc-1.5.02-200705172019.sh.piece_as
  edrc-1.5.02-200705172019.sh.piece_at
  edrc-1.5.02-200705172019.sh.piece_au
  edrc-1.5.02-200705172019.sh.piece.sh
done.

```

done.

5) list all defined applications

See also **pack.cfg(4)** for the description of how to define applications.

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][bash]: ./pack -l
```

pack - create application package, by Chr. Walther

| APPLICATION | PREFIX | TS_PREFIX | DESCRIPTION |
|-----------------|-----------|-----------|------------------------------------|
| default | edrc_WA2L | edrc | WA2L/edrc complete |
| edrc | edrc_WA2L | edrc | WA2L/edrc complete |
| edrc_CORE | edrc_WA2L | edrc | WA2L/edrc core files (without cust |
| edrc_NOSCRIPTS | edrc_WA2L | edrc | WA2L/edrc without customer recover |
| edrc_NOSECURITY | edrc_WA2L | edrc | WA2L/edrc without security files |
| edrc_UPDATE | edrc_WA2L | edrc | WA2L/edrc without recovery script |

(6)

| TS_PREFIX | RELEASE | TIMESTAMP_DATE |
|-----------|---------|------------------|
| edrc | 1.4.06 | 0818125404 |
| edrc | 1.4.07 | 1120173804 |
| edrc | 1.4.08 | 2005-06-27 09:36 |
| edrc | 1.5.00 | 2005-07-14 14:20 |
| edrc | 1.5.01 | 2007-02-27 16:16 |
| edrc | 1.5.02 | 2007-12-14 13:00 |
| edrc | 1.5.03 | 2007-12-18 12:42 |
| edrc | 1.5.04 | 2008-04-09 17:13 |
| edrc | 1.5.05 | 2008-04-30 13:13 |
| edrc | 1.5.06 | 2008-05-09 08:06 |
| edrc | 1.5.07 | 2008-05-23 23:40 |

(11)

6) create a package containing the core files only

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][bash]: ./pack -t -a edrc_CORE
```

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...

application information ...

APPLICATION : edrc_CORE

APPLICATION_PREFIX ... : edrc

APPLICATION_NAME : WA2L/edrc

APPLICATION_RELEASE .. : 1.5.02

DESCRIPTION : WA2L/edrc core files (without customer s

done.

package information ...

format : shar

type : RELEASE

```
file ..... : /tmp/edrc-1.5.02-200705172024.sh
done.
write timestamp ...(/opt/edrc/var/pack/ts/edrc-1.5.02)... done.
write sadm file ...(/opt/edrc/var/pack/sadm/edrc-1.5.02-200705172024.
evaluate files to be packed ...(4201 files)... done.
pack files to package file ...(15222.7 kBytes)... done.
done.
```

SEE ALSO

edrcintro(1), **edrcsetup**(1m), **pack.cfg**(4), **patchinstall**(1m), **stat**(3)

NOTES

pack is used to create the shell archive of WA2L/edrc .

BUGS

-

AUTHOR

pack was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pack.cfg – configuration file for pack

SYNOPSIS

edrc/etc/pack.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **pack** command.

FILEFORMAT

The format for options is **OPTION=VALUE** .

The format of a application definition is:

```
APPLICATION_name(){
    OPTION1=VALUE
    OPTION2=VALUE
    :
    OPTIONn=VALUE
}
```

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION**. If you like to use default settings simply do not specify a *VALUE*.

Normally there are multiple application settings defined in one configuration file.

The setting **APPLICATION_default** must be defined in a configuration file.

OPTIONS**GLOBAL OPTIONS**

This options influence the general behavior of the **pack** command.

LOG Log output dir of **pack**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

APPLICATION OPTIONS

A application definition has the following structure:

```
APPLICATION_name(){
    PACKAGE_NAME="text"
    DESCRIPTION="text"
    APPLICATION_PREFIX="text"
    APPLICATION_RELEASE="version_number"
    TIMESTAMP_PREFIX="text"
    APPLICATION_BASEDIR="directory"
    APPLICATION_COMPONENTS="directory"
    OUTPUT_DIRECTORY="directory"
    EXCLUDE_LIST="regex_list"
    CONFIGFILE_LIST="regex_list"
}
```

The options recognized in a application definition are:

name Name of the application. The name defined here has to be specified in the **-a** option in the **pack** command. The application with the name *default* is called when the **-a** option is not specified when using the **pack** command. Therefore each **pack.cfg** file should at least define a profile with the name *default*.

PACKAGE_NAME

Name of the software package. This normally correlates to the package name in the documentation/man pages.

Example: PACKAGE_NAME="WA2L/edrc"

Default: PACKAGE_NAME="?"

DESCRIPTION

Application description.

Example: DESCRIPTION="WA2L/edrc complete"

Default: DESCRIPTION=""

APPLICATION_PREFIX

This is the prefix of the application package, as:
<APPLICATION_PREFIX>-1.5.01-20070306.sh.

Example: APPLICATION_PREFIX="edrc"

Default: APPLICATION_PREFIX="myApp"

APPLICATION_RELEASE

Release number (version) of the application software.

Example: APPLICATION_RELEASE=""^`aproot`/sbin/edrc -V^`"

Default: APPLICATION_RELEASE=<has to be specified interactively>

TIMESTAMP_PREFIX

This is the prefix of the application timestamp, as:

<TIMESTAMP_PREFIX>-1.5.01. Normally the timestamp prefix should be set to <APPLICATION_PREFIX>

Example: TIMESTAMP_PREFIX="edrc"

Default: TIMESTAMP_PREFIX=""<APPLICATION_PREFIX>"

APPLICATION_BASEDIR

Directory where the software is currently installed.

Example: APPLICATION_BASEDIR=""^`aproot^`"

Default: APPLICATION_BASEDIR=""^`aproot^`"

PACKAGE_COMPONENTS

Directory in **edrc/var/pack**/**<PACKAGE_COMPONENTS>** where the additional components used for the generation of the different package formats is located. See section **FILES** in **pack**(1m) for more information.

Example: PACKAGE_COMPONENTS="edrc"

Default: PACKAGE_COMPONENTS=""<APPLICATION_PREFIX>

OUTPUT_DIRECTORY

Existing directory where the software package file is created. This directory should have at least 1.5 times the space available as the resulting software file size.

Example: OUTPUT_DIRECTORY="/dat/sw/apps"

Default: OUTPUT_DIRECTORY="/tmp"

EXCLUDE_LIST

List (regex) of files to be excluded from the software package. To verify the exclude definition invoke **pack -L** to list the files that would be packed into the software package.

Example: EXCLUDE_LIST="/.sav/ ^edrc/var/log/.+.log\$"

Default: EXCLUDE_LIST=""

CONFIGFILE_LIST

List (regex) of config files of the software to be packed. This setting is for package formats that support the special handling of configuration files. This are the formats **rpm**, **depot** and **pkg**. The **shar** format currently does not treat config files special. But for future enhancements the **CONFIGFILE_LIST** should be defined, too.

Example: `CONFIGFILE_LIST="^edrc/etc/."`

Default: `CONFIGFILE_LIST=""`

VARIABLES

Currently no official variables are provided to be used in application settings.

To avoid defining the same settings multiple times it is allowed to chain application definitions by "calling" another definition at the beginning of an application definition. In fact you should do this at least in the **APPLICATION_default** definition. Therefore it is allowed to use the application setting variables to enhance a definition. This makes especially sense to enhance an **EXCLUDE_LIST** by first including another definition and then using the **\$EXCLUDE_LIST** variable in the definition of the **EXCLUDE_LIST**. See also definitions in the `edrc/etc/pack.cfg` file and the **EXAMPLE** section for examples for the method of chaining definitions.

EXAMPLES**1) example application definition for the WA2L/edrc application**

See the "call" of **APPLICATION_edrc** in the **APPLICATION_default** and **APPLICATION_edrc_NOSECURITY** definition to chain the definitions to save definition effort.

```
# edrc: WA2L/edrc complete
#
APPLICATION_edrc() {
    PACKAGE_NAME="WA2L/edrc"
    DESCRIPTION="WA2L/edrc complete"
    APPLICATION_PREFIX="edrc"
    APPLICATION_RELEASE="\`approot\`/sbin/edrc -V"
    APPLICATION_BASEDIR="\`approot"
    PACKAGE_COMPONENTS="edrc"
    OUTPUT_DIRECTORY="/tmp"
    EXCLUDE_LIST="\
        /\ .sav/ \
        /\ .\[ ^ssh\] \
        /\ .*\ .swp\$ \
        ^edrc/etc/.*\ .2[0-9]{7}\$ \
        ^edrc/bin/.*\ .2[0-9]{7}\$ \
        ^edrc/doc/.man/.+\.ps\$ \
        ^edrc/doc/.man/.+\.pdf\$ \
        ^edrc/doc/.man/.+\.html\$ \
        ^edrc/lib/.*\ .2[0-9]{7}\$ \
        ^edrc/lib/edrcpack/ \
        ^edrc/src/.+\.o\$ \
        ^edrc/var/backup/.+ \
        ^edrc/var/barbedwire/cache/.+ \
        ^edrc/var/connection/cache/.+ \
    "
```

```

^edrc/var/contrib/edrc/.+ \
^edrc/var/edrcpack/ \
^edrc/var/lock/.+\.lock$ \
^edrc/var/log/.+\.log$ \
^edrc/var/repl/.+ \
^edrc/var/sfi/app_mgmt/applications/.+ \
^edrc/var/sfi/csdocs/hosts/data/.+ \
^edrc/var/sfi/db/.+ \
^edrc/var/settings/.+ \
^edrc/var/shell/sh_history\..* \
^edrc/var/shell/sh_var\..* \
^edrc/var/shell/edrc_history\..* \
^edrc/var/passwdsyncd/.+ \
^edrc/var/pwsync/ \
"
CONFIGFILE_LIST="\
^edrc/etc/.+ \
"
}

# edrc_NOSECURITY: WA2L/edrc without security files
#
APPLICATION_edrc_NOSECURITY() {
    APPLICATION_edrc
    PACKAGE_NAME="WA2L/edrc"
    DESCRIPTION="WA2L/edrc without security files"
    APPLICATION_PREFIX="edrc"
    APPLICATION_RELEASE="\`approot\`/sbin/edrc -V"
    APPLICATION_BASEDIR="\`approot \`"
    PACKAGE_COMPONENTS="edrc"
    OUTPUT_DIRECTORY="/tmp"
    EXCLUDE_LIST="\
    $EXCLUDE_LIST \
    ^edrc/var/connection/security/edrc/+/+/+/ \
    ^edrc/var/connection/security/edrc/.ssh/ \
    "
    CONFIGFILE_LIST="\
    ^edrc/etc/.+ \
    "
}

# default: Default setting. This setting is effective when pack is
# execute without options.
#
APPLICATION_default() {
    APPLICATION_edrc
}

```

SEE ALSO

edrcintro(1), **pack**(1).

NOTES

-

BUGS

-

AUTHOR

pack.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

passwdcombine – combine users into a passwd file

SYNOPSIS

edrc/bin/passwdcombine [**-h**]

passwdcombine -t *systype* **-s** *save_file*

passwdcombine -t *passwd* **-f** '*file_1 select_1*' { **-f** '*file_n select_n*' } **-o** *outfile*

passwdcombine -t *trusted* **-f** '*file_1 select_1*' { **-f** '*file_n select_n*' } **-o** *outdir*

AVAILABILITY

WA2L/edrc

DESCRIPTION

combine password files or password databases.

This is useful if you have to apply a new set of users to a system and you like to prevent a change of certain users (as system users).

OPTIONS

-h this usage message.

-t *system_type*

type of system:

passwd /etc/passwd contains all password

trusted HP Trusted System

-s *save_file*

save file (without suffix)

-f '*file_n select_n*'

file_n source file

select_n select file

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|--|
| 0 | no error. |
| 1 | file specified in -f option does not exist. |
| 1 | /etc/passwd or /tcb/ does not exist. |
| 2 | operating system not supported. |
| 2 | cannot write to output file. |
| 3 | output file already exists. |
| 4 | usage printed. |
| 5 | the passwdcombine command was aborted. |

FILES

| | |
|--------------------|--|
| /etc/passwd | password file. |
| /tcb/ | password database on a trusted HP-UX system. |

EXAMPLES

-

SEE ALSO

edrcintro(1), **passwd(5)**

NOTES

-

BUGS

-

AUTHOR

passwdcombine was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

passwdsort – filter to sort

SYNOPSIS

edrc/bin/passwdsort [**-h**]

cat */etc/passwd* | **passwdsort**

passwdsort < */etc/passwd*

AVAILABILITY

WA2L/edrc

DESCRIPTION

sort a **passwd** file or data stream from **stdin** by UID.

OPTIONS

-h usage message.

ENVIRONMENT

-

EXIT STATUS

4 usage printed.

0 no error.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **passwd(4)**

NOTES

-

BUGS

-

AUTHOR

passwdsort was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

passwdsyncd – password synchronization daemon

SYNOPSIS

edrc/bin/passwdsyncd [**-h**]

passwdsyncd -s

passwdsyncd -m { on | off }

AVAILABILITY

WA2L/edrc

DESCRIPTION

With the **passwdsyncd** it is possible to synchronize the password database over a list of systems without the need of using NIS or NIS+.

To run the **passwdsyncd** it is necessary to have the times synchronized between all participating systems.

The **passwdsyncd** has the following features implemented:

- no master server needed, all participating systems are equivalent. Therefore this solution is also preferred to use in a clustered environment where it should be ensured that all nodes are identical.
- the **passwdsyncd** can be started at any time, it will not overwrite newer password entries on participating systems.
- the daemon can be switched centrally into a maintenance mode where all synchronization between the participating systems is on hold.
- if configured, the synchronization peers can grow dynamically without the need of reconfiguration of the other systems already participating in a synchronization.
- if configured, user lockouts are not synchronized.
- the synchronization data is submitted encrypted between the participating systems.

STATES

The **passwdsyncd** has the following states which are also recorded to the logfile:

check check if there are password database entries newer then the last password synchronization attempt of the local system. After startup this is the first state.

- synch** synchronize password database entries to remote system(s).
- wait** sleep between checks. This interval can be configured with the **SYNC_INTERVAL** setting in the **passwdsyncd.cfg** file.
- maintenance_mode_on**
promote the maintenance mode to the systems participating in a synchronization.
- maintenance_mode_off**
clear the maintenance mode on all systems participating in a synchronization.
- maintenance**
the **passwdsyncd** is in maintenance mode. During maintenance mode all password synchronization is on hold.
- stop** the **passwdsyncd** has been stopped with the **-s** option.
- abort** the **passwdsyncd** has been killed with the **kill** command. This method of stopping the **passwdsyncd** is not recommended.

OPTIONS

- h** usage message.
- m** maintenance mode control:
- on** switch maintenance mode on.
- off** switch maintenance mode off.
- s** stop a running **passwdsyncd**.

SIGNALS

The following signals are handled by **passwdsyncd**. Do not use other signals as those listed below, as long as you do not really know what you are doing and what the consequences are. In general there is no need to invoke those signals by your own, this signals are used for inter process communication of the **passwdsyncd**.

- INT** interrupt from keyboard (Ctrl+C). Issuing this signal, the daemon will go to the **abort** state.
- TERM** kill the process without an argument. Issuing this signal, the daemon will go to the **abort** state.
- HUP** stop the **passwdsyncd**. This signal is issued when **passwdsyncd -s** is issued and will put the **passwdsyncd** into the **stop** state.

- USR1** set the **passwdsyncd** into maintenance mode. The state of **passwdsyncd** after receiving this signal is **maintenance**.
- USR2** clear maintenance mode of **passwdsyncd**. The state of **passwdsyncd** after receiving this signal is **check**.

EXIT STATUS

- 0** no error.
- 1** daemon started as the wrong operating system user.
- 2** operating system is not supported. See **osid(3)** if you get this error.
- 3** cannot write to the lockfile.
- 4** usage listed.
- 5** there is already a **passwdsyncd** instance running on the system.
- 6** configuration file **edrc/etc/passwdsyncd.cfg** does not exist.
- 8** cannot write to the logfile.
- 9** no permission to stop the daemon.
- 11** a temporary directory could not be claimed or created in **/var/tmp**. Check the system temporary directory **/var/tmp** if you get this error, it is an indicator of system intrusion.
- 12** no permission to handle daemon maintenance mode.
- 13** system security type not supported.

FILES

- edrc/etc/passwdsyncd.cfg**
configuration file for **passwdsyncd**.
- edrc/var/log/**
default logfile location of **passwdsyncd**.
- edrc/var/passwdsyncd/**
default spool location of **passwdsyncd** to save password information.

edrc/var/lock/

default lock directory of **passwdsyncd**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **passwdsyncd.cfg**(4), **osid**(3), **passwdsyncd_apply**(3), **remote_copy**(3), **remote_shell**(3)

NOTES

currently **passwdsyncd** supports the synchronization of HP-UX trusted systems.

BUGS

-

AUTHOR

passwdsyncd was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

passwdsyncd.cfg – configuration file for passwdsyncd

SYNOPSIS

edrc/etc/passwdsyncd.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **passwdsyncd(1m)** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**START_USER**

This is the user **passwdsyncd** has to be started with.

Example: **START_USER=eroot**

Default: **START_USER=root**

SYSTEM_TYPE

Type of system the **passwdsyncd** is started on. Set **SYSTEM_TYPE** to *trusted* for HP-UX Trusted system, to *passwd* for systems with passwords in **/etc/passwd** (not implemented yet) and to *shadow* for systems with passwords in **/etc/shadow** (not implemented yet). It probably won't be possible to mix system types.

Example: **SYSTEM_TYPE=trusted**

Default: **SYSTEM_TYPE=trusted**

LOG

Log output dir of **passwdsyncd**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: **LOG=/var/opt/edrc/log**

Default: **LOG=var/log**

SPOOL_DIR

Spool output basedir to save the transferred password information. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation. In general it is not recommended to set the **SPOOL_DIR** within WA2L/edrc, locate it in a system own directory.

Example: SPOOL_DIR=/var/spool

Default: SPOOL_DIR=var

SYNC_HOSTLIST

Space separated list of hosts that take part on a password synchronization.

Example: SYNC_HOSTLIST="dcdbsi61 dcdbsi62 dcdbsi71"

Default: SYNC_HOSTLIST=

DYNAMIC_SYNC_EXPANSION

Expand the **SYNC_HOSTLIST** dynamically with hosts from which additional synchronizations has been received. If this option is set to *True* dynamic expansion is enabled, if it is set to *False* it is disabled.

Example: DYNAMIC_SYNC_EXPANSION=True

Default: DYNAMIC_SYNC_EXPANSION=False

LOCKOUT_SYNC

Synchronize locked users. This setting has effect on the following system types: trusted. If **LOCKOUT_SYNC** is set to *locked* users are synchronized, if set to *False* a lock of a user will remain on one system and will not be synchronized to other systems.

Example: LOCKOUT_SYNC=True

Default: LOCKOUT_SYNC=False

DIST_MODE

Comma separated list of modes used to distribute the password information. The supported modes are: *rtools* which result in the use of **r**cp and **r**sh or *OpenSSH* which results in the use of **s**cp and **s**sh for distribution. If a comma separated list is provided, the connection initiation is made in the sequence specified. A pseudo distribution mode is *default* which results in the use of the **CONNECTION_MODE** specified in the configuration files **remote_shell.cfg** and **remote_copy.cfg**. It is not allowed to specify *default* as part of a comma separated list.

Example: DIST_MODE=rtools,OpenSSH

Default: DIST_MODE=rtools

DIST_USER

User used to distribute the files in **SCRIPTS_BASEDIR**. The home of this user is considered as the root of the WA2L/edrc installation. See **edrcsetup**(1m) for information about user settings needed by WA2L/edrc.

Example: DIST_USER=edrc

Default: DIST_USER=edrc

EDRC_OWNER

Owner of the WA2L/edrc software. This is the user the EDRC environment is installed with. See **edrcsetup**(1m) for information about user settings needed by WA2L/edrc.

Example: EDRC_OWNER=root

Default: EDRC_OWNER=root

SYNC_INTERVAL

Synchronization interval in seconds. **passwdsyncd** checks in the interval specified here if passwords have changed since the last check. If so a synchronization attempt to the other hosts participating in a synchronization will take place. The **SYNC_INTERVAL** should be set to a figure that the passwords are synchronized within a "reasonable" timeframe. Due to the fact that **passwdsyncd** is not a Master-Slave construct, the more servers participate in a synchronization, the more synchronization traffic will be generated.

Example: SYNC_INTERVAL=300

Default: SYNC_INTERVAL=600

EXCLUDE_USERS

Exclude this users from synchronization.

Example: EXCLUDE_USERS=root,bin,daemon,sys,adm,uucp,lp,hpdb,sshd

Default: EXCLUDE_USERS=""

LOCKDIR

Lock dir of **passwdsyncd**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation. In general it is not recommended to set the lockdir within WA2L/edrc, locate it in a system own directory.

Example: LOCKDIR=/var/run

Default: LOCKDIR=var/lock

SECRET Synchronization secret password. This password has to be identical on all hosts which participate in a synchronization.

Example: SECRET="myT0PsecretPa55w0rd"

Default: SECRET="wBnCYP4HOb8Xw"

SECRET_LIFETIME

Synchronization secret (**SECRET**) lifetime in 100 seconds (example: **SECRET_LIFETIME** = 1 equals to 100 seconds). After this duration the encrypted synchronization file will not be accepted by the hosts which participate in a synchronization. To disable secret ageing, set **SECRET_LIFETIME** to 0.

This setting is currently partly ignored: the secret lifetime is 100 seconds independent of the

settings made here, except it is set to 0.

Example: SECRET_LIFETIME=1

Default: SECRET_LIFETIME=1

ENCRYPTION_MODE

Mode of encryption of the transferred password information. Currently the following modes are supported: *Enigma* (simple Unix crypt. This option requires **crypt**(1) to be installed on all participating systems), *AES* (Advanced Encryption Standard. This option requires **ccrypt**(1) to be installed on all participating systems) and *NO* (no encryption. This option is not recommended for permanent solutions, but does not need **crypt**(1) or **ccrypt**(1)).

Example: ENCRYPTION_MODE=AES

Default: ENCRYPTION_MODE=Enigma

SEE ALSO

crypt(1), **ccrypt**(1), **edrcintro**(1), **passwdsyncd**(1m)

NOTES

ccrypt is distributed with WA2L/edrc.

BUGS

-

AUTHOR

passwdsyncd.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

passwdsyncd_apply – read a long line from stdin

SYNOPSIS

edrc/lib/passwdsyncd_apply [**-h**]

passwdsyncd_apply -s remote_system

AVAILABILITY

WA2L/edrc

DESCRIPTION

apply passwords distributed with **passwdsyncd** to the target system.

This command is called from the **passwdsyncd**.

OPTIONS

-h usage message.

-s remote_system
system to apply the password information to.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 1 | command is started with the wrong operating system user. |
| 2 | operating system not supported. |
| 3 | spool directory does not exist. |
| 4 | usage displayed. |
| 6 | config file not found. |

- 7 synchronization attempt to local system.
- 8 cannot write to log file.
- 10 daemon not running.
- 11 system is in maintenance mode.
- 13 system type not supported yet.
- 14 wrong system type on remote system.

FILES

edrc/etc/passwdsyncd.cfg

configuration file for **passwdsyncd**(1m) and **passwdsyncd_apply**(4).

edrc/var/passwdsyncd/

spool directory.

edrc/var/lock/

lock directory.

EXAMPLES

-

SEE ALSO

edrcintro(1), **passwdsyncd**(1m), **passwdsyncd.cfg**(4), **remote_copy**(3), **remote_shell**(3)

NOTES

passwdsyncd_apply is used internally by the **passwdsyncd**(1m) command.

BUGS

-

AUTHOR

passwdsyncd_apply was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

patchinstall – patch a WA2L/edrc installation

SYNOPSIS

edrc/lib/patchinstall [**-h**]

patchinstall [**-n** | **-m**] **-f** *patch_file*

patchinstall **-i** | **-I** [*number*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

patch a WA2L/edrc installation with a *patch_file* generated by **pack**(1m).

patchinstall also cleans up corpses of previous application installations. Corpses are files and/or directories that were part of the application, but are no longer used, are out of date (as: the .PDF manual pages of older versions in the **edrc/doc/** directory or older versions of installed binaries) or are no longer needed due to design changes in the application.

In the default configuration the files to be patched and the identified corpses are backedup prior to the installation/removal of the files.

The backup behaviour, the logfile detail depth and other settings are configurable in the configuration file **patchinstall.cfg** that is read prior to the installation of the files that are part of the *patch_file*.

When **patchinstall** is invoked, the following steps are performed:

- 1.) print system information. This includes information as the *patch_file* to be installed, if **patchinstall** is invoked with the **-n** option, and other information about the system where the patch will be applied.
- 2.) print the patch meta data. This information is also available in the directory **edrc/var/pack/sadm/** for each installed patch.
- 3.) backup the files residing in the application directory to be patched. In the active configuration file **patchinstall.cfg** can be configured if the files to be patched should be backedup or not.
- 4.) purge backedup patched files. The number of kept backup file versions (generations) and if the purging of the backup files should take place at all can be configured in the **patchinstall.cfg** configuration file.
- 5.) install the patch files.

- 6.) update timestamps of symlinks that are part of the patch. This feature is not available on all operating systems. On systems where changing of symlink timestamps is not possible, the symlink will have the timestamp of the patch installation.
- 7.) evaluate corpses in the current installation. The corpses are evaluated on each system. There is no master file list used in a patch or the application to identify corpses. This makes it possible, that own added files are not rated as a corpse. If there are doubts if a certain file would be rated as a corpse by a patch to be installed, it is possible to run **patchinstall** in the NOEXECUTE mode (**patchinstall -n -f patch_file**), where no actions are performed, but all messages are printed to stdout.
- 8.) backup files that were identified as corpses. In the active configuration file **patchinstall.cfg** can be configured if the files identified as corpses should be backed up or not previous to the removal.
- 9.) purge backed up corpses. The number of kept backup file versions (generations) and if the purging of the backup files should take place at all can be configured in the **patchinstall.cfg** configuration file.
- 10.) remove the corpses from the application installation.
- 11.) re-run step 5.) and 6.) if step 5.) terminated with errors.

The backups are written to the backup directory in the ***.cpio.gz** format, that can be viewed using the **lscomp(1)**, **llcomp(1)** and **catcomp(1)** commands.

OPTIONS

- h** usage message.
- n** NOEXECUTE. This option causes that no actions are performed, but all messages are printed. This can be used to check the consequences of a patch installation.
- m** only print meta-data of the patch. This option can also be used to verify if the file is really an WA2L/edrc patch-file.
- l** only list the files contained in the patch.
- f patch_file**
patch file to be installed. The *patch_file* has to be specified with an absolute path.
- i** list installed patches based on the **patchinstall** log file.
- I** print last installed patch in record format.
- number* number of most recent patches to list; default is **1**,

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|---|
| 0 | no error. |
| 1 | <i>patch_file</i> does not exist. |
| 2 | error evaluating the files that are part of the WA2L/edrc installation. |
| 3 | error evaluating the static files of the WA2L/edrc installation. |
| 4 | usage listed. |
| 5 | sadm file not found in <i>patch_file</i> , this indicates, that the <i>patch_file</i> is not a WA2L/edrc patch. |
| 6 | configuration file patchinstall.cfg does not exist. |
| 7 | command aborted, as pressing Ctrl+C. |

FILES

edrc/etc/patchinstall.cfg
configuration file for **patchinstall**.

EXAMPLES

1) remove corpses of an installation

If you want to clean up corpses on an existing installation without installing a WA2L/edrc patch of a newer version, create a patch to the current version you have installed and then install this patch you just created on the same system:

```
[ /root ]
[ root@acme001 ][*edrc*/bash]: pack -p `~edrc/sbin/edrc -V`
```

pack - create application package, by Chr. Walther

```
create package of application in '/opt/edrc' ...
application information ...
  APPLICATION ..... : default
  APPLICATION_PREFIX ... : edrc_WA2L
  APPLICATION_NAME .... : WA2L/edrc
  APPLICATION_RELEASE .. : 1.5.10
  DESCRIPTION ..... : WA2L/edrc complete
done.
```

```

package information ...
  format ..... : shar
  type ..... : PATCH (to version 1.5.10)
  file ..... : /tmp/edrc_WA2L-1.5.10-200809071349-PATCH
done.
write sadm file ...(/opt/edrc/var/pack/sadm/edrc_WA2L-1.5.10-20080907
evaluate files to be packed ...(9 files)... done.
evaluate properties of files to be packed ... done.
pack files to package file ...(6.98535 kBytes)... done.
done.

[ /root ]
[ root@acme001 ][*edrc*/bash]: patchinstall \
                                -f /tmp/edrc_WA2L-1.5.10-200809071349-P
patchinstall - patch a WAL2/edrc installation, by Chr. Walther

install patch to application in '/opt/edrc' ...
  system information ...
    :
    :
  done.
  patch meta data ...
    :
    :
  done.
  backup files to be patched ...
    :
    :
  done.
  purge backedup patched files ...
    :
    :
  done.
  install patch files ...
    :
    :
  done.
  update symlink timestamps ...
    :
    :
  done.
  backup corpses to be removed ...
    :
    :
  done.
  purge backedup corpses ...
    :
    :
  done.
  remove corpses ...
    :
    :
  done.

```

done.

SEE ALSO

edrcintro(1), **edrcsetup**(1m), **catcomp**(1), **lscomp**(1), **llcomp**(1), **pack**(1m), **patchinstall.cfg**(4), **stat**(3)

NOTES

Due to the fact, that an installation of a *patch_file* using **patchinstall** does not cause additional command invocations (as: pre- or post-exec scripts), a patch can be applied multiple times to an application installation without side effects.

BUGS

-

AUTHOR

patchinstall was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

patchinstall.cfg – configuration file for patchinstall

SYNOPSIS

edrc/etc/patchinstall.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **patchinstall** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**START_USER**

This is the user **patchinstall** has to be started with.

Example: START_USER=root

Default: START_USER=root

BACKUP_DIR

patchinstall saves all files ahead of patching the files to a backup file to this directory.

Example: BACKUP_DIR=/var/opt/edrc/backup

Default: BACKUP_DIR=var/backup

LOG

Log output dir of **patchinstall**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

LOG_LEVEL

Level of information written to the logfile. If all messages should be logged to the logfile, set this setting to *verbose*, if only the minimum should be logged, set it to *minimal*. The *verbose* setting causes that all output as seen on the screen (stdout) will also be saved to the logfile.

Example: LOG_LEVEL=minimal

Default: LOG_LEVEL=verbose

PACKAGE_FILES_APP

Application files.

Example: PACKAGE_FILES_APP=edrc

Default: PACKAGE_FILES_APP=""

STATIC_FILES_APP

Static installed files.

Example: STATIC_FILES_APP=edrc_STATIC

Default: STATIC_FILES_APP=""

BACKUP_PATCHED_FILES

If set to *True*, the files that will be patched are backed up previous to overwriting. Else set it to *False*.

Example: BACKUP_PATCHED_FILES=False

Default: BACKUP_PATCHED_FILES=True

BACKUP_PATCHED_FILES_GENERATIONS

When this setting is set to a value other than 0, purging of old patched files backup files takes place. The number specified here is the number of most recent backup files that are kept.

Example: BACKUP_PATCHED_FILES_GENERATIONS=10

Default: BACKUP_PATCHED_FILES_GENERATIONS=0

BACKUP_CORPSES

If set to *True*, the identified corpses are backed up previous to removal. Else set it to *False*.

Example: BACKUP_CORPSES=False

Default: BACKUP_CORPSES=True

BACKUP_CORPSES_GENERATIONS

When this setting is set to a value other than 0, purging of old corpses backup files takes place. The number specified here is the number of most recent backup files that are kept.

Example: BACKUP_CORPSES_GENERATIONS=10

Default: BACKUP_CORPSES_GENERATIONS=0

SEE ALSO

patchinstall(1m), **edrcintro**(1)

NOTES

-

BUGS

-

AUTHOR

patchinstall.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pdfmetaedit – GUI to edit PDF meta data

SYNOPSIS

edrc/bin/pdfmetaedit

AVAILABILITY

WA2L/edrc

DESCRIPTION

GUI to edit meta data of a PDF file.

See also <http://zaro.github.io/pdf-metadata-editor/>.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **epub2pdf(1)**, **pdfscissors(1)**, <http://zaro.github.io/pdf-metadata-editor/>

NOTES

This command was developed by "zarro".

BUGS

-

AUTHOR

pdfmetaedit was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pdfscissors – GUI to crop PDF file

SYNOPSIS

edrc/bin/pdfscissors

AVAILABILITY

WA2L/edrc

DESCRIPTION

GUI to crop PDF file for eBook reader.

This command was developed by Abdullah Al Mazed (Gagan).

See also <http://pdfscissors.com/>.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **pdfmetaedit(1)**, **<http://pdfscissors.com/>**

NOTES

-

BUGS

-

AUTHOR

pdfscissors was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

perlenv – print environment needed to start perl scripts

SYNOPSIS

edrc/lib/perlenv [-n]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print the environment used by **perl** to access the Perl modules bundled with WA2L/edrc.

To set the environment prior to the execution of **perl**, invoke:

```
eval `perlenv`
```

This command is **only** to be used to prepare the environment to execute Perl scripts in "Recovery Script" trees. See **perl_modules(3)** for information regarding this usage.

When writing Perl scripts for WA2L/edrc, do **not** use **perlenv**, start the script thru the **.perl_wrapper**. See **perl_wrapper(1)** for information about integrating a Perl script into WA2L/edrc.

OPTIONS

-n no **export** *VARIABLE* ... output.

ENVIRONMENT**\$PERLLIB**

this variable holds Perl module locations additional to the standard applying to an installation. This variable will be expanded with the bundled Perl module locations when executing the **perlenv** command.

\$LD_LIBRARY_PATH

path of additional shared libraries loaded by the **ld** dynamic linker. This variable will be expanded with the shared library locations for a certain Os-Id distributed with WA2L/edrc when executing the **perlenv** command.

\$SHLIB_PATH

this variable is expanded as the **\$LD_LIBRARY_PATH** is done. This variable has basically the same function as the **\$LD_LIBRARY_PATH** and is mostly active on HP/UX systems.

\$PATH command search path. This variable will be expanded with the path where the **perl** interpreter was found on the system when executing the **perlenv** command. See also **perl_wrapper(1)** for more information.

EXIT STATUS

0 always

FILES

edrc/lib/perl_wrapper.cfg
configuration file for **.perl_wrapper** and **perlenv**.

edrc/lib/perl/pm/

location of all bundled perl modules. The related directories in **edrc/lib/perl/pm/** are prepended to the **\$PERLLIB** environment variable when executing **perlenv**. The **pmdesc -h** command can also be used to display the current Perl library path.

The directory structure in **edrc/lib/perl/pm/** is:

Legacy Structure:

```
perl<rel>/<vers>/
perl<rel>/site_perl/<vers>/
perl<rel>/vendor_perl/<vers>/
perl<rel>/edrc_perl/
perl<rel>/share/<vers>/bin/
perl<rel>/share/<vers>/man/
perl<rel>/share/site_perl/<vers>/bin/
perl<rel>/share/site_perl/<vers>/man/
perl<rel>/share/vendor_perl/<vers>/bin/
perl<rel>/share/vendor_perl/<vers>/man/
```

New Structure:

```
perl<rel>/<vers>/root/
perl<rel>/edrc_perl/
```

Where **<rel>** is the major version number (e.g. 5) and **<vers>** is the whole perl version (e.g. 5.14.3).

The "Legacy Structure" and the "New Structure" can be and is mixed. However, the "Legacy Structure" will not disappear and all files will remain as they are; new modules will be installed into the more simple "New Structure". When using the **.perl_wrapper** to start a **perl** script and the **perlenv** commands to initialize the **perl** environment, scripts need no change.

See also: **perl_modules(3)**, **pmdesc(1)**.

edrc/src/perlmodules/

source packages of all bundled perl modules to enable a fast compilation exercise when having a different perl version on the system. See **README** file and use the provided **Makefile(s)** to

build and install the related module.

edrc/lib/<Os-Id>/libs/

location of operating system dependent shared libraries outside of Perl. This directory is appended to the **\$LD_LIBRARY_PATH** and the **\$SHLIB_PATH** environment variables when executing **perlenv**.

EXAMPLES

1) Start a Perl script from a recovery script

To start a Perl script that uses a bundled Perl module from a recovery script, set the Perl library path (**@INC**) using the **'eval 'perlenv''** command in the recovery script and then call the **perl** interpreter.

Recovery script:

```
#!/bin/ksh
#
# 1:ascript - A Recovery Script
#
# [00] 08.02.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x
:
:
eval `perlenv`
perl ./perlscript
:
```

Perl script:

```
#
# perlscript - Perl script using bundled DBI module
#
# [00] 18.03.2009 CWa Initial Version
#
:
:
use DBI;
:
```

2) Start a Perl script from a recovery script using the **_env file mechanism**

Create an **_env** file (using the **env edrc** command) and start the Perl script from the recovery script.

_env file:

```
#
# _env - Environment settings for commands in /apps/eg
#
# [00] 18.03.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

eval `perlenv`
:
```

Recovery script:

```
#!/bin/ksh
#
# 1:ascript - A Recovery Script
#
# [00] 08.02.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x
:
:
perl ./perlscript
:
```

Perl script:

```
#
# perlscript - Perl script using bundled DBI module
#
# [00] 18.03.2009 CWa Initial Version
#
:
:
use DBI;
:
```

3) Write a recovery script in Perl

Create an **_env** file (using the **env edrc** command) and write the recovery script in Perl.

_env file:

```
#
# _env - Environment settings for commands in /apps/eg
#
# [00] 18.03.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

eval `perlenv`
```

:

Perl recovery script:

```
#!/usr/bin/env perl
#
# 1:ascrip - A Recovery Script
#
# [00] 08.02.2009 CWa Initial Version
#
:
:
use DBI;
:
```

Note the use of `'#!/usr/bin/env perl'` in the magic key to start the Perl script. This ensures that the **perl** interpreter is found on the system, even when it is not installed in the default location you will normally specify. The path to the **perl** interpreter is added to the **\$PATH** using the **perlenv** command.

4) Write a perl script outside of WA2L/edrc and profit from bundled perl modules

The startup method method (see also: <https://perldoc.perl.org/perlrun>) below allows to start the **perl** interpreter from wherever it is installed and also enables to use all **perl** modules bundled with WA2L/edrc using the **perlenv** command to dynamically initialize the environment.

This method is similar to the use of the **.perl_wrapper**, but the **perl** script can be placed somewhere of your liking outside of WA2L/edrc.

Set the permissions of the script to executable:

```
chmod +x perlscrip
```

Perl script:

```
#!/bin/ksh
#
# perlscrip - Perl script using bundled modules
#
# [00] 18.03.2009 CWa Initial Version
#
eval `~edrc/lib/perlenv`
#! -*- perl -*- -p
eval 'exec perl -x -SS $0 ${1+"$@"}'
if 0;

use strict;
use warnings;
use File::Basename;
use REST::Client;
use JSON;
use DBI;
use Data::Dumper;
```

```
use Getopt::Std;
use WA2L::Util;
:
:
```

5) Test the availability of a specific perl module

Set the needed **perl** environment:

```
[ / ]
[ root@acme007 ][*edrc*/bash]: eval `perlenv`
```

Try to access the **perl** module:

```
[ / ]
[ root@acme007 ][*edrc*/bash]: perl -e "use REST::Client;"
```

If an error message like

```
Can't locate REST/Client.pm in @INC (you may need to install the REST::
(@INC contains: /etc/perl ... /usr/lib/x86_64-linux-gnu/perl-base) at -
BEGIN failed--compilation aborted at -e line 1.
```

appears, the module cannot be found, if the module can be found there is no output.

SEE ALSO

edrcintro(1), **cpanm(1)**, **ld(1)**, **osid(3)**, **perl(1)**, **perlversion(3)**, **pmdesc(1)**, **perl_modules(3)**, **perl_wrapper(1)**,
http://search.cpan.org/~mschwern/ExtUtils-MakeMaker/lib/ExtUtils/MakeMaker.pm#make_install, http://search.cpan.org/~jesse/perl/INSTALL#Coexistence_with_earlier_versions_of_perl_5, <https://perldoc.perl.org/perlrun>

NOTES

The following excellent description of the directory structure is a partial extract of http://search.cpan.org/~jesse/perl-5.17.8/INSTALL#Coexistence_with_earlier_versions_of_perl_5 which has been written by Jesse Vincent <esse+cpan@fsck.com>.

Be aware, that the **perlenv** and **.perl_wrapper** commands are a bit less strict in the resolution of the directories listed in the **\$PERLLIB** environment variable.

Coexistence with earlier versions of perl 5

Perl 5.14 is not binary compatible with earlier versions of Perl. In other words, you will have to recompile your XS modules.

In general, you can usually safely upgrade from one version of Perl (e.g. 5.X.Y) to another similar minor version (e.g. 5.X.(Y+1))) without re-compiling all of your extensions. You can also safely leave the old version around in case the new version causes you problems for some reason.

Usually, most extensions will probably not need to be recompiled to be used with a newer version of Perl.

Here is how it is supposed to work. (These examples assume you accept all the Configure defaults.)

Suppose you already have version 5.8.7 installed. The directories searched by 5.8.7 are typically like:

```
/usr/local/lib/perl5/5.8.7/$archname
/usr/local/lib/perl5/5.8.7
/usr/local/lib/perl5/site_perl/5.8.7/$archname
/usr/local/lib/perl5/site_perl/5.8.7
```

Now, suppose you install version 5.8.8. The directories searched by version 5.8.8 will be:

```
/usr/local/lib/perl5/5.8.8/$archname
/usr/local/lib/perl5/5.8.8
/usr/local/lib/perl5/site_perl/5.8.8/$archname
/usr/local/lib/perl5/site_perl/5.8.8

/usr/local/lib/perl5/site_perl/5.8.7/$archname
/usr/local/lib/perl5/site_perl/5.8.7
/usr/local/lib/perl5/site_perl/
```

Notice the last three entries -- Perl understands the default structure of the \$sitelib directories and will look back in older, compatible directories. This way, modules installed under 5.8.7 will continue to be usable by 5.8.7 but will also be accessible to 5.8.8. Further, suppose that you upgrade a module to one which requires features present only in 5.8.8. That new module will get installed into **/usr/local/lib/perl5/site_perl/5.8.8** and will be available to 5.8.8, but will not interfere with the 5.8.7 version.

The last entry, **/usr/local/lib/perl5/site_perl/**, is there so that 5.6.0 and above will look for 5.004-era pure perl modules.

Lastly, suppose you now install 5.10.0, which is not binary compatible with 5.8.x. The directories searched by 5.10.0 (if you don't change the Configure defaults) will be:

```
/usr/local/lib/perl5/5.10.0/$archname
/usr/local/lib/perl5/5.10.0
/usr/local/lib/perl5/site_perl/5.10.0/$archname
/usr/local/lib/perl5/site_perl/5.10.0

/usr/local/lib/perl5/site_perl/5.8.8

/usr/local/lib/perl5/site_perl/5.8.7

/usr/local/lib/perl5/site_perl/
```

Note that the earlier \$archname entries are now gone, but pure perl modules from earlier versions will still be found.

This way, you can choose to share compatible extensions, but also upgrade to a newer version of an extension that may be incompatible with earlier versions, without breaking the earlier versions' installations.

BUGS

-

AUTHOR

perlenv was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

perlpod – the Plain Old Documentation format

DESCRIPTION

Pod is a simple-to-use markup language used for writing documentation for Perl, Perl programs, and Perl modules.

Translators are available for converting Pod to various formats like plain text, HTML, man pages, and more.

Pod markup consists of three basic kinds of paragraphs: ordinary, verbatim, and command.

Ordinary Paragraph

Most paragraphs in your documentation will be ordinary blocks of text, like this one. You can simply type in your text without any markup whatsoever, and with just a blank line before and after. When it gets formatted, it will undergo minimal formatting, like being rewrapped, probably put into a proportionally spaced font, and maybe even justified.

You can use formatting codes in ordinary paragraphs, for **bold**, *italic*, `code-style`, hyperlinks, and more. Such codes are explained in the "Formatting Codes" section, below.

Verbatim Paragraph

Verbatim paragraphs are usually used for presenting a codeblock or other text which does not require any special parsing or formatting, and which shouldn't be wrapped.

A verbatim paragraph is distinguished by having its first character be a space or a tab. (And commonly, all its lines begin with spaces and/or tabs.) It should be reproduced exactly, with tabs assumed to be on 8-column boundaries. There are no special formatting codes, so you can't italicize or anything like that. A \ means \, and nothing else.

Command Paragraph

A command paragraph is used for special treatment of whole chunks of text, usually as headings or parts of lists.

All command paragraphs (which are typically only one line long) start with "=", followed by an identifier, followed by arbitrary text that the command can use however it pleases. Currently recognized commands are

```
=pod
=head1 Heading Text
=head2 Heading Text
=head3 Heading Text
=head4 Heading Text
=over indentlevel
=item stuff
=back
=begin format
=end format
=for format text...
=encoding type
=cut
```

To explain them each in detail:

```
=head1 Heading Text
=head2 Heading Text
=head3 Heading Text
=head4 Heading Text
```

Head1 through head4 produce headings, head1 being the highest level. The text in the rest of this paragraph is the content of the heading. For example:

```
=head2 Object Attributes
```

The text "Object Attributes" comprises the heading there. The text in these heading commands can

use formatting codes, as seen here:

```
=head2 Possible Values for C<$/>
```

Such commands are explained in the "Formatting Codes" section, below.

```
=over indentlevel
=item stuff...
=back
```

Item, over, and back require a little more explanation: “=over” starts a region specifically for the generation of a list using “=item” commands, or for indenting (groups of) normal paragraphs. At the end of your list, use “=back” to end it. The *indentlevel* option to “=over” indicates how far over to indent, generally in ems (where one em is the width of an “M” in the document’s base font) or roughly comparable units; if there is no *indentlevel* option, it defaults to four. (And some formatters may just ignore whatever *indentlevel* you provide.) In the *stuff* in *=item stuff...*, you may use formatting codes, as seen here:

```
=item Using C<$|> to Control Buffering
```

Such commands are explained in the "Formatting Codes" section, below.

Note also that there are some basic rules to using “=over” ... “=back” regions:

- Don’t use “=item”s outside of an “=over” ... “=back” region.
- The first thing after the “=over” command should be an “=item”, unless there aren’t going to be any items at all in this “=over” ... “=back” region.
- Don’t put “=head n ” commands inside an “=over” ... “=back” region.
- And perhaps most importantly, keep the items consistent: either use “=item *” for all of them, to produce bullets; or use “=item 1.”, “=item 2.”, etc., to produce numbered lists; or use “=item foo”, “=item bar”, etc.—namely, things that look nothing like bullets or numbers.

If you start with bullets or numbers, stick with them, as formatters use the first “=item” type to decide how to format the list.

```
=cut
```

To end a Pod block, use a blank line, then a line beginning with “=cut”, and a blank line after it. This lets Perl (and the Pod formatter) know that this is where Perl code is resuming. (The blank line before the “=cut” is not technically necessary, but many older Pod processors require it.)

```
=pod
```

The “=pod” command by itself doesn’t do much of anything, but it signals to Perl (and Pod formatters) that a Pod block starts here. A Pod block starts with *any* command paragraph, so a “=pod” command is usually used just when you want to start a Pod block with an ordinary paragraph or a verbatim paragraph. For example:

```
=item stuff()
```

```
This function does stuff.
```

```
=cut
```

```
sub stuff {
    ...
}
```

```
=pod
```

Remember to check its return value, as in:


```
=for comment
```

Make sure that all the available options are documented!

Some *formatnames* will require a leading colon (as in "`=for :formatname`", or "`=begin :formatname`" ... "`=end :formatname`"), to signal that the text is not raw data, but instead *is* Pod text (i.e., possibly containing formatting codes) that's just not for normal formatting (e.g., may not be a normal-use paragraph, but might be for formatting as a footnote).

```
=encoding encodingname
```

This command is used for declaring the encoding of a document. Most users won't need this; but if your encoding isn't US-ASCII, then put a `=encoding encodingname` command very early in the document so that pod formatters will know how to decode the document. For *encodingname*, use a name recognized by the Encode::Supported module. Some pod formatters may try to guess between a Latin-1 or CP-1252 versus UTF-8 encoding, but they may guess wrong. It's best to be explicit if you use anything besides strict ASCII. Examples:

```
=encoding latin1
```

```
=encoding utf8
```

```
=encoding koi8-r
```

```
=encoding ShiftJIS
```

```
=encoding big5
```

`=encoding` affects the whole document, and must occur only once.

And don't forget, all commands but `=encoding` last up until the end of its *paragraph*, not its line. So in the examples below, you can see that every command needs the blank line after it, to end its paragraph. (And some older Pod translators may require the `=encoding` line to have a following blank line as well, even though it should be legal to omit.)

Some examples of lists include:

```
=over
```

```
=item *
```

```
First item
```

```
=item *
```

```
Second item
```

```
=back
```

```
=over
```

```
=item Foo()
```

```
Description of Foo function
```

```
=item Bar()
```

```
Description of Bar function
```

```
=back
```

Formatting Codes

In ordinary paragraphs and in some command paragraphs, various formatting codes (a.k.a. “interior sequences”) can be used:

I<text> — italic text

Used for emphasis (“be I<careful!>”) and parameters (“redo I<LABEL>”)

B<text> — bold text

Used for switches (“perl's B<-n> switch”), programs (“some systems provide a B<chfn> for that”), emphasis (“be B<careful!>”), and so on (“and that feature is known as B<autovivification>”).

C<code> — code text

Renders code in a typewriter font, or gives some other indication that this represents program text (“C<gmtime (\$^T)>”) or some other form of computerese (“C<drwxr-xr-x>”).

L<name> — a hyperlink

There are various syntaxes, listed below. In the syntaxes given, text, name, and section cannot contain the characters '/' and '|'; and any '<' or '>' should be matched.

- L<name>
Link to a Perl manual page (e.g., L<Net::Ping>). Note that name should not contain spaces. This syntax is also occasionally used for references to Unix man pages, as in L<crontab(5)>.
- L<name/"sec"> or L<name/sec>
Link to a section in other manual page. E.g., L<perlsyn/"For Loops">
- L</"sec"> or L</sec>
Link to a section in this manual page. E.g., L</"Object Methods">

A section is started by the named heading or item. For example, L<perlvar/\$.> or L<perlvar/"\$."> both link to the section started by “=item \$.” in perlvar. And L<perlsyn/For Loops> or L<perlsyn/"For Loops"> both link to the section started by “=head2 For Loops” in perlsyn.

To control what text is used for display, you use “L<text | ...>”, as in:

- L<text | name>
Link this text to that manual page. E.g., L<Perl Error Messages|perldiag>
- L<text | name/"sec"> or L<text | name/sec>
Link this text to that section in that manual page. E.g., L<postfix "if"|perlsyn/"Statement Modifiers">
- L<text | /"sec"> or L<text | /sec> or L<text | "sec">
Link this text to that section in this manual page. E.g., L<the various attributes|/"Member Data">

Or you can link to a web page:

- L<scheme:...>
L<text | scheme:...>
Links to an absolute URL. For example, L<http://www.perl.org/> or L<The Perl Home Page|http://www.perl.org/>.

E<escape> — a character escape

Very similar to HTML/XML &foo; “entity references”:

- E<lt> — a literal < (less than)

- E<gt> — a literal > (greater than)
- E<verbar> — a literal | (*vertical bar*)
- E<sol> — a literal / (*solidus*)

The above four are optional except in other formatting codes, notably L<...>, and when preceded by a capital letter.

- E<htmlname>

Some non-numeric HTML entity name, such as E<eacute>, meaning the same thing as é in HTML — i.e., a lowercase e with an acute (/–shaped) accent.

- E<number>

The ASCII/Latin-1/Unicode character with that number. A leading “0x” means that *number* is hex, as in E<0x201E>. A leading “0” means that *number* is octal, as in E<075>. Otherwise *number* is interpreted as being in decimal, as in E<181>.

Note that older Pod formatters might not recognize octal or hex numeric escapes, and that many formatters cannot reliably render characters above 255. (Some formatters may even have to use compromised renderings of Latin-1/CP-1252 characters, like rendering E<eacute> as just a plain “e”.)

F<filename> — used for filenames

Typically displayed in italics. Example: "F<.cshrc>"

S<text> — text contains non-breaking spaces

This means that the words in *text* should not be broken across lines. Example: S<\$x ? \$y : \$z>.

X<topic name> — an index entry

This is ignored by most formatters, but some may use it for building indexes. It always renders as empty-string. Example: X<absolutizing relative URLs>

Z<> — a null (zero-effect) formatting code

This is rarely used. It’s one way to get around using an E<...> code sometimes. For example, instead of "NE<lt>3" (for "N<3" you could write "NZ<><3" (the "Z<>" breaks up the "N" and the "<" so they can’t be considered the part of a (fictitious) "N<...>" code).

Most of the time, you will need only a single set of angle brackets to delimit the beginning and end of formatting codes. However, sometimes you will want to put a real right angle bracket (a greater-than sign, '>') inside of a formatting code. This is particularly common when using a formatting code to provide a different font-type for a snippet of code. As with all things in Perl, there is more than one way to do it. One way is to simply escape the closing bracket using an E code:

```
C<$a E<lt>=>E<gt> $b>
```

This will produce: "\$a <=> \$b"

A more readable, and perhaps more “plain” way is to use an alternate set of delimiters that doesn’t require a single “>” to be escaped. Doubled angle brackets (“<<” and “>>”) may be used *if and only if there is whitespace right after the opening delimiter and whitespace right before the closing delimiter!* For example, the following will do the trick:

```
C<< $a <=> $b >>
```

In fact, you can use as many repeated angle-brackets as you like so long as you have the same number of them in the opening and closing delimiters, and make sure that whitespace immediately follows the last '<' of the opening delimiter, and immediately precedes the first '>' of the closing delimiter. (The whitespace is ignored.) So the following will also work:

```
C<<< $a <=> $b >>>
C<<<< $a <=> $b >>>>
```

And they all mean exactly the same as this:


```
C<$a E<lt>=E<gt> $b>
```

The multiple-bracket form does not affect the interpretation of the contents of the formatting code, only how it must end. That means that the examples above are also exactly the same as this:

```
C<< $a E<lt>=E<gt> $b >>
```

As a further example, this means that if you wanted to put these bits of code in C (code) style:

```
open(X, ">>thing.dat") || die $!
$foo->bar();
```

you could do it like so:

```
C<<< open(X, ">>thing.dat") || die $! >>>
C<< $foo->bar(); >>
```

which is presumably easier to read than the old way:

```
C<open(X, "E<gt>E<gt>thing.dat") || die $!>
C<$foo-E<gt>bar();>
```

This is currently supported by `pod2text` (`Pod::Text`), `pod2man` (`Pod::Man`), and any other `pod2xxx` or `Pod::Xxxx` translators that use `Pod::Parser 1.093` or later, or `Pod::Tree 1.02` or later.

The Intent

The intent is simplicity of use, not power of expression. Paragraphs look like paragraphs (block format), so that they stand out visually, and so that I could run them through `fmt` easily to reformat them (that's F7 in my version of **vi**, or Esc Q in my version of **emacs**). I wanted the translator to always leave the ' and ` and " quotes alone, in verbatim mode, so I could slurp in a working program, shift it over four spaces, and have it print out, er, verbatim. And presumably in a monospace font.

The Pod format is not necessarily sufficient for writing a book. Pod is just meant to be an idiot-proof common source for `nroff`, HTML, TeX, and other markup languages, as used for online documentation. Translators exist for **pod2text**, **pod2html**, **pod2man** (that's for *nroff*(1) and *troff*(1)), **pod2latex**, and **pod2fm**. Various others are available in CPAN.

Embedding Pods in Perl Modules

You can embed Pod documentation in your Perl modules and scripts. Start your documentation with an empty line, a “=head1” command at the beginning, and end it with a “=cut” command and an empty line. The **perl** executable will ignore the Pod text. You can place a Pod statement where **perl** expects the beginning of a new statement, but not within a statement, as that would result in an error. See any of the supplied library modules for examples.

If you're going to put your Pod at the end of the file, and you're using an `__END__` or `__DATA__` cut mark, make sure to put an empty line there before the first Pod command.

```
__END__
```

```
=head1 NAME
```

```
Time::Local - efficiently compute time from local and GMT time
```

Without that empty line before the “=head1”, many translators wouldn't have recognized the “=head1” as starting a Pod block.

Hints for Writing Pod

-

The **podchecker** command is provided for checking Pod syntax for errors and warnings. For example, it checks for completely blank lines in Pod blocks and for unknown commands and formatting codes. You should still also pass your document through one or more translators and proofread the result, or print out the result and proofread that. Some of the problems found may be bugs in the translators, which you may or may not wish to work around.

- If you're more familiar with writing in HTML than with writing in Pod, you can try your hand at writing documentation in simple HTML, and converting it to Pod with the experimental Pod::HTML2Pod module, (available in CPAN), and looking at the resulting code. The experimental Pod::PXML module in CPAN might also be useful.
- Many older Pod translators require the lines before every Pod command and after every Pod command (including “=cut”) to be a blank line. Having something like this:

```
# - - - - -
=item $firecracker->boom()
```

```
This noisily detonates the firecracker object.
```

```
=cut
sub boom {
    ...
```

...will make such Pod translators completely fail to see the Pod block at all.

Instead, have it like this:

```
# - - - - -
```

```
=item $firecracker->boom()
```

```
This noisily detonates the firecracker object.
```

```
=cut
```

```
sub boom {
    ...
```

- Some older Pod translators require paragraphs (including command paragraphs like “=head2 Functions”) to be separated by *completely* empty lines. If you have an apparently empty line with some spaces on it, this might not count as a separator for those translators, and that could cause odd formatting.
- Older translators might add wording around an L<> link, so that L<Foo::Bar> may become “the Foo::Bar manpage”, for example. So you shouldn't write things like the L<foo> documentation, if you want the translated document to read sensibly. Instead, write the L<Foo::Bar|Foo::Bar> documentation or L<the Foo::Bar documentation|Foo::Bar>, to control how the link comes out.
- Going past the 70th column in a verbatim block might be ungracefully wrapped by some formatters.

SEE ALSO

perlpodspec, “PODs: Embedded Documentation” in perlsyn, perlnewmod, perldoc, pod2html, pod2man, podchecker.

AUTHOR

Larry Wall, Sean M. Burke

NAME

perl_modules – list of WA2L/edrc bundled Perl module man pages

SYNOPSIS

-

AVAILABILITY

WA2L/edrc

DESCRIPTION

This man page lists man pages of the main Perl modules that are part of WA2L/edrc package and distributed with it.

Many Perl modules need other modules to work. All modules that were identified to be needed by a certain Perl module bundled with WA2L/edrc are also included into the package.

To start a Perl script that uses a bundled Perl module from a recovery script, set the Perl library path (@INC) using the **'eval 'perlenv''** command in the recovery script and then call the **perl** interpreter:

```
#!/bin/ksh
#
# 1:ascript - A Recovery Script
#
# [00] 08.02.2009 CWa    Initial Version
#
#
test "$DEBUG" = True && set -x
:
:
eval `perlenv`
perl ./perlscript
:
```

An other possibility is to create an **_env** file (using the **env** edrc command) with the content:

```
#
# _env - Environment settings for commands in /apps/eg
#
# [00] 18.03.2009 CWa    Initial Version
#
#
test "$DEBUG" = True && set -x

eval `perlenv`
:
```

and use the bundled Perl module using the '**use module**;' clause:

```
#!/usr/bin/env perl
#
# perlscript - Perl script using bundled DBI module
#
# [00] 18.03.2009 CWa    Initial Version
#
:
:
use DBI;
:
```

With this 2nd method it is also possible to write recovery scripts in Perl that use the bundled Perl modules.

Note the use of '**#!/usr/bin/env perl**' in the magic key to start the Perl script. This ensures that the **perl** interpreter is found on the system, even when it is not installed in the default location you will normally specify.

PERL MODULE MAN PAGE REFERENCE OF WA2L/edrc

CGI(3pm) Simple Common Gateway Interface Class.

CMS::MediaWiki(3pm) Perl extension for creating, reading and updating MediaWiki pages.

DBD::SQLite(3pm) Self Contained RDBMS in a DBI Driver.

DBI(3pm) Database independent interface for Perl.

File::Tail(3pm) Perl extension for reading from continuously updated files.

HTML::Parser(3pm) HTML parser class.

MediaWiki::API(3pm) Perl interface to the MediaWiki API.

MIME::Base64(3pm) Encoding and decoding of base64 strings

MIME::Lite(3pm) Low-calorie MIME generator.

Text::vCard(3pm) a package to edit and create a single vCard (RFC 2426).

Tk(3pm) a graphical user interface toolkit for Perl.

WA2L::Util(3) General WA2L/edrc Utility Functions.

COMPILE/INSTALL BUNDLED WA2L/edrc Perl MODULES

Because often Perl modules also need to be compiled there is a good chance that on certain operating system versions this modules have to be compiled.

If you experience missing modules when using a WA2L/edrc Perl command (watch for PSCR in **compatibility(1)**) or in an own Perl script using bundled Perl modules you might be forced to compile it for your system.

To do so:

0) goto shell in edrc or start shell

The following steps have to be executed in the **shell** due to the fact that the install mechanism relay on the environment provided by the **shell** command.

```
host-001# ~edrc/bin/shell
```

1) goto source directory

All bundled Perl modules are included in WA2L/edrc as source packages to enable a quick and efficient compilation process on future systems.

```
[ /root ]
[ root@host-001 ][*eshell*/bash]: cdsrc; cd perlmodules

[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]:
```

2) list install sequence

Because often modules relay on other modules, the given sequence below helps to a successful compilation and installation of modules.

```
[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: make list

#
# LIST MODULES
#

:
Compress::Compress-Raw-Zlib
Compress::Compress-Zlib
LWP::LWP-UserAgent
JSON::JSON
URI::URI
MediaWiki::MediaWiki-API
File::File-Tail
CMS::CMS-MediaWiki
:
```

Identify the module(s) that has been reported as missing on the output list. Often a whole chain of modules are missing, where the above sequence gives help on the compile/install order.

3) compile module

The provided **Makefile** handles all steps to compile, test and install a bundled Perl module into the WA2L/edrc structure and not to change the underlying operating system. This steps make

the related Perl module "portable" and on similar systems the modules don't have to be compiled they can simply be distributed to it using the **pack(1m)** and **patchinstall(1m)** commands of WA2L/edrc.

```
[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]:

[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: make help

[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: make build
```

To build a subset of the whole module list defined in the **ALL** variable in the **Makefile**, add the module to the **TST** variable and invoke:

```
[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: make buildtst
```

4) install module into WA2L/edrc

Install the module into the WA2L/edrc structure into the **edrc/lib/perl/pm/** directory without to change the underlying operating system.

```
[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: make install
```

5) clean up files (optional)

Clean up all produced files during the compile/test/install process. This is an optional step.

```
[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: make clean
```

6) create a patch release of your installation (optional)

If you compiled all missing bundled Perl modules you can create a PATCH release of the WA2L/edrc package to install the produced files on all your other systems.

```
[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: pack -d ~edrc/var/sw -a edrc_UPDATE -

pack - create application package, by Chr. Walther

create package of application in '/opt/edrc' ...
application information ...
APPLICATION ..... : edrc_UPDATE
APPLICATION_PREFIX ... : edrc_WA2L
APPLICATION_NAME ..... : WA2L/edrc
APPLICATION_RELEASE .. : 1.5.46
DESCRIPTION ..... : WA2L/edrc without recovery script-, co
done.
package information ...
format ..... : shar
type ..... : PATCH (to version 1.5.46)
file ..... : host-001:/opt/edrc/var/sw/edrc_WA2L-1.
```

```

done.
write sadm file ...(/opt/edrc/var/pack/sadm/edrc_WA2L-1.5.46-201408
evaluate files to be packed ... (729 files)... done.
evaluate properties of files to be packed ... done.
pack files to package file ... (12917 kB)... done.
done.

```

7) install the patch on all your other systems

The most convenient way to patch all your systems is to use the **edrcupgrade** contributed command in **edrc** :

```

[ /opt/edrc/src/perlmodules ]
[ root@host-001 ][*eshell*/bash]: sat

sat@host-001> edrcupgrade
(s)tart (d)ownload only (i)nstall only (c)ancel [c] :i

DEPLOY AND INSTALL AN WA2L/edrc PATCH

patchfile basedir [/opt/edrc/var/sw]:

list downloaded patch files? <ync> [y] :y

AVAILABLE EDRC PATCH files:

FROM      TO      FILENAME
-----
1.5.46    1.5.46    edrc_WA2L-1.5.46-201408231741-PATCH-1.5.46.cpio.gz
(1)

patchfile to install : edrc_WA2L-1.5.46-201408231741-PATCH-1.5.46.cpio.

:

```

You can also install the generated patch more manually first distributing the patch file using **filedist(1)** and then installing it with the **patchinstall(1m)** command on all systems using the **rcmd(1)** command.

ENVIRONMENT

-

EXIT STATUS

-

FILES

edrc/lib/perl/pm/<perl_release>/<perl_version>/

location of all bundled Perl modules. This directory is automatically pre-pended to the **@INC** array when the Perl script is started thru the **.perl_wrapper**. See also: **perl_wrapper(1)**.

edrc/src/perlmodules/

source packages of all bundled Perl modules.

EXAMPLES

-

SEE ALSO

edrcintro(1), **contrib.edrcupgrade(1m)**, **filedist(1)**, **lspm(1)**, **make(1)**, **pack(1m)**, **patchinstall(1m)**, **perl_wrapper(1)**, **perl(1)**, **perlenv(3)**, **pmdesc(1)**, **rcmd(1)**, **CGI(3pm)**, **CMS::MediaWiki(3pm)**, **DBD::SQLite(3pm)**, **DBI(3pm)**, **File::Tail(3pm)**, **HTML::Parser(3pm)**, **MediaWiki::API(3pm)**, **MIME::Base64(3pm)**, **MIME::Lite(3pm)**, **Tk(3pm)**, **URI(3pm)**, **WA2LUtil(3)**, **WA2L::Util(3)**

NOTES

When Perl scripts are started thru the **perl_wrapper** it is not needed to use the **perlenv** command or to set the perl **@INC** array somehow; simply put the perl script into the **lib/perl/** directory and create the symbolic link having the identical name as the Perl script file in **bin/** or **lib/** pointing to the **.perl_wrapper**.

See also **perl_wrapper(1)** for examples of integrating Perl scripts into WA2L/edrc.

BUGS

-

AUTHOR

perl_modules was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

perlversion – print version (major.minor.patch) of Perl interpreter

SYNOPSIS

edrc/lib/perlversion

AVAILABILITY

WA2L/edrc

DESCRIPTION

print **perl** version of **perl** interpreter found on the system.

See also **perl_wrapper(3)** for additional information.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **appreversion(3)**, **perlenv(3)**, **perl_wrapper(3)**, **revision(3)**, **scriptrevision(3)**

NOTES

-

BUGS

-

AUTHOR

perlversion was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

perl_wrapper – wrap Perl scripts to ensure perl startup

SYNOPSIS

edrc/bin/perlscript -> **.perl_wrapper**

edrc/lib/perlscript -> **.perl_wrapper**

AVAILABILITY

WA2L/edrc

DESCRIPTION

Wrap Perl scripts to avoid using *#!/usr/bin/perl* or similar path names in the magic key. This enhances the portability of WA2L/edrc commands written in Perl to systems where **perl** is not installed in the default install locations.

perl has to be installed in **/bin/**, **/sbin/**, **/usr/local/bin/** or **/usr/contrib/bin/**.

Further **perl** locations can be configured in the config file **perl_wrapper.cfg**.

The Perl modules bundled with WA2L/edrc that are installed in **lib/perl/pm/<perl-revision>/** are included in the Perl library path (**@INC**) and the **ld** environment variables (**\$SHLIB_PATH**, **\$LD_LIBRARY_PATH**) are expanded to the directory **lib/<OS-Id>/libs/** that might contain additional needed shared libraries.

To start a new command thru the **.perl_wrapper**, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ][*edrc*/bash]: ln -s .perl_wrapper new_cmd
```

- 2.) install the Perl script in the **edrc/lib/perl/** directory:

```
[ /opt/edrc/lib/perl ]
[ root@acme001 ][*edrc*/bash]: vi new_cmd

[ /opt/edrc/lib/perl ]
[ root@acme001 ][*edrc*/bash]: chmod 644 new_cmd
```

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- 101** shell (**perl**) not found. If **perl** is installed on the system and you get this error, add the **perl** location in the configuration file **perl_wrapper.cfg**.
- 102** the Perl script to be started that should be located in **lib/perl/** does not exist.
- 103** user calling the command has no permission to access/execute the called *perlscript*.
- 106** configuration file **perl_wrapper.cfg** does not exist.
- 107** the **.perl_wrapper** was called directly.

FILES

- etc/perl_wrapper.cfg**
configuration file for the **.perl_wrapper** command.
- lib/perl/** location of the wrapped Perl scripts. This files should have the file permissions *644* to show, that those scripts should not be started directly.
- lib/perl/pm/<perl-revision>/**
location of the perl modules bundled with WA2L/edrc.
- lib/<OS-Id>/libs/**
locations of the provided shared libraries for a certain operating system (**OS-Id**).

EXAMPLES

-

SEE ALSO

edrcintro(1), **binprobe(1m)**, **cmdlist(1m)**, **daemon_wrapper(1)**, **java_wrapper(1)**, **ksh_wrapper(1)**, **lua_wrapper(1)**, **ln(1)**, **ld(1)**, **osid(3)**, **os_wrapper(1)**, **perl(1)**, **perlenv(3)**, **perlversion(3)**, **perl_modules(3)**, **perl_wrapper.cfg(4)**, **pmdesc(1)**, **shlib(3)**

NOTES

-

BUGS

-

AUTHOR

perl_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

perl_wrapper.cfg – configuration file for .perl_wrapper

SYNOPSIS

edrc/etc/perl_wrapper.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **.perl_wrapper** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SEARCH_PATH**

Colon separated search path where to search for the **perl** interpreter.

Example: SEARCH_PATH="/bin:/usr/bin:/sbin:/opt/perl/bin:/usr/local/bin:/usr/contrib/bin"

Default: SEARCH_PATH="/bin:/usr/bin:/sbin:/opt/perl/bin:/usr/local/bin:/usr/contrib/bin"

MODULE_DIR

Perl Modules contributed with WA2L/edrc.

Example: MODULE_DIR=lib/perl/pm

Default: MODULE_DIR=lib/perl/pm

SEE ALSO

edrcintro(1), **perl_wrapper**(1)

NOTES

-

BUGS

-

AUTHOR

perl_wrapper.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pf_wrapper – start command thru sudo or RBAC if user is privileged

SYNOPSIS

edrc/pbin/edrc_command -> .pf_wrapper

AVAILABILITY

WA2L/edrc

DESCRIPTION

Wrap WA2L/edrc commands to be started thru **pfexec**(1) or **sudo**(8).

The attributes of the privileged *edrc_command* is done with the original operating system commands (**sudo**, **pfexec**) to execute it with elevated permissions.

The **pf_wrapper**(1) does simplify the startup of those commands. The user only needs to add the **~edrc/pbin/** directory into the **\$PATH** environment variable and does not need to know if a system (as Solaris or HP-UX) is configured to use RBAC (Role Based Access Control) **pfexec** to execute a command with elevated permissions or **sudo**.

To include an additional WA2L/edrc command into this mechanism, create a symlink to the **.pf_wrapper** in the **edrc/pbin/** directory.

If a system uses both startup mechanisms (RBAC and **sudo**), first it is checked if a command is configured to be started thru RBAC, then **sudo**.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

107 direct call of **.pf_wrapper**.

126 permission denied to start a command. This error occurs, if the WA2L/edrc command is not configured to be executed thru **sudo** or **RBAC**.

127 command does not exist in WA2L/edrc (**edrc/sbin/**, **edrc/bin/** or **edrc/lib/**).

x exit status of the related command.

FILES

bin/ binary directory in WA2L/edrc where the *edrc_command* is located.

lib/ library directory in WA2L/edrc where the *edrc_command* is located.

pbins/ binary directory for commands that could be started with elevated permissions. This directory only contains symlinks to the **.pf_wrapper**.

sbin/ binary directory where the **edrc(1m)** command is located.

var/log/pf_wrapper.log
log file for the **.pf_wrapper** command.

/etc/sudoers
typical configuration file for **sudo(8)**.

/etc/security/exec_attr
execution profiles database for RBAC.

/etc/security/prof_attr
profile description database for RBAC.

/etc/user_attr
extended user attributes database for RBAC.

EXAMPLES

1) sudo configuration example

To start some commands thru **sudo**, the following lines could be added to the **/etc/sudoers** file, using the **visudo** command. To ensure, that the user has to supply the own password to execute the privileged commands, the settings in line 55 and 56 should be commented out or removed from the file.

```
[ /etc ]
[ root@host-001 ][*eshell*/bash]: sav sudoers; visudo

1  #
2  # /etc/sudoers - sudo access definition file
3  #
4  # [01] 25.02.2012 CWa  +*_EDRC_* definitions
5  #
6  ##
```

```

7  ## Sudoers allows particular users to run various commands
8  ## as the root user, without needing the root password.
9  ##
:
18 ## User Aliases
19 User_Alias    ROLE_EDRC_ADM = john, fred
20 User_Alias    ROLE_EDRC_OPS = barney
21 User_Alias    ROLE_EDRC_USR = wilma, betty
22
23 # WA2L/edrc definitions
24 #
25 ROLE_EDRC_ADM ALL = PASSWD: CMD_EDRC_USR, CMD_EDRC_OPS, CMD_EDRC
26 ROLE_EDRC_OPS ALL = PASSWD: CMD_EDRC_USR, CMD_EDRC_OPS
27 ROLE_EDRC_USR ALL = PASSWD: CMD_EDRC_USR
28 Cmnd_Alias    CMD_EDRC_ADM = /opt/edrc/bin/shell, /opt/edrc/bin/
29                                     /opt/edrc/sbin/edrc, /opt/edrc/bin/
30 Cmnd_Alias    CMD_EDRC_OPS = /opt/edrc/bin/osup
31 Cmnd_Alias    CMD_EDRC_USR = /opt/edrc/bin/asup, /opt/edrc/bin/
:
51 # In the default (unconfigured) configuration, sudo asks for the
52 # password. This allows use of an ordinary user account
53 # administration of a fresh installed system. When configuring s
54 # delete the two following lines:
55 #Defaults     targetpw      # specify passwd of target user i.e
56 #ALL          ALL=(ALL) ALL # only together with 'Defaults targe
:
~
~
~

```

The concept of the configuration example above is, that all commands that are to be used by an user are defined in the command alias **CMD_EDRC_USR**, the additional commands that are used by operational personnel are defined in the command alias **CMD_EDRC_OPS** and the additional commands that are used by administrators are listed in **CMD_EDRC_ADM**.

To build roles, the role **ROLE_EDRC_USR** consists of the **CMD_EDRC_USR** commands, the **ROLE_EDRC_OPS** consists of the **CMD_EDRC_USR** and **CMD_EDRC_OPS** commands and the **ROLE_EDRC_ADM** consists of all three **CMD_EDRC_*** command sets (**CMD_EDRC_USR**, **CMD_EDRC_OPS** **CMD_EDRC_ADM**).

Finally a role is connected to a real user using the **User_Alias** directive. In a more automated environment you most likely don't want to edit the **sudoers** file every time a change in your user base occurs, therefore a **User_Alias** can be connected to an operating system group (*%groupname*) with the setting:

```

:
18 ## User Aliases
19 User_Alias    ROLE_EDRC_ADM = %edrc_adm
20 User_Alias    ROLE_EDRC_OPS = %edrc_ops
21 User_Alias    ROLE_EDRC_USR = %edrc_usr
22
:

```

In this example the operating system groups **edrc_adm**, **edrc_ops** and **edrc_usr** represent the roles defined in the **sudoers** file.

2) RBAC configuration example

-

3) add ~edrc/pbin directory to users ~/.bashrc for Bourne again Shell

```
[ /home/fred ]
[ fred@host-001 ][bash]: vi ~/.bashrc
:
27 # Add edrc/pbin to $PATH
28 PATH=~edrc/pbin:$PATH; export PATH
~
~
```

4) add ~edrc/pbin directory to users ~/.kshrc for Korn Shell

```
[ /home/fred ]
[ fred@host-001 ][bash]: vi ~/.kshrc
:
27 # Add edrc/pbin to $PATH
28 PATH=~edrc/pbin:$PATH; export PATH
~
~
```

SEE ALSO

edrcintro(1), **edrcsetup**(1m), **sudo**(8), **sudoers**(5), **uniqupath**(3), **visudo**(8), **profiles**(1), **exec_attr**(4), **user_attr**(4), **prof_attr**(4)

NOTES

Currently the WA2L/edrc commands **asup**(1), **edrc**(1m), **lotsctl**(1), **osup**(1), **psup**(1), **sat**(1), **shell**(1) and **sys**(1) are available in the **pbin/** directory.

Be aware, that always the original command (for example: **~edrc/bin/shell**) has to be configured in **sudo** or RBAC and not the **pbin/.pf_wrapper** command.

Therefore, if you followed the suggestions how to configure **sudo**(8) in the **edrcsetup**(1m) manual page, you don't have to change anything in the **sudo** configuration.

BUGS

-

AUTHOR

pf_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pip – package manager for Python packages in WA2L/edrc

SYNOPSIS

edrc/bin/pip [*options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

start the **pip** command **pip3**(1) with settings tuned for the WA2L/edrc package and the Python3 virtual environment (**venv**) distributed with the package.

See also **pip3**(1) for additional information.

OPTIONS

options all options of the **pip** command, see: **pip3**(1).

ENVIRONMENT**\$VIRTUAL_ENV**

is set to the **edrc/lib/python/pym/<OSID>/<major>.<minor>/** directory.

\$PYTHONPATH

is set to **\$VIRTUAL_ENV/lib/python<major>.<minor>/site-packages;\$PYTHONPATH**.

\$PATH

is set to **\$VIRTUAL_ENV/bin;\$PATH**.

\$PIP_DESTINATION_DIRECTORY

if not set, points to the **edrc/var/sw/** directory if write access is possible.

\$PIP_LOG_FILE

is set to **edrc/var/log/pip.log** if writeable, else to **/dev/null**.

EXIT STATUS

x exit status of **pip3**(1).

FILES

edrc/var/log/pip.log
log file of **pip**.

edrc/var/sw/
software directory for Python3 package downloads.

edrc/lib/python/pym/<OSID>/<major>.<minor>/
Python3 virtual environment (venv) distributed with WA2L/edrc.

edrc/lib/python/pym/<OSID>/<major>.<minor>/bin/pip
Python3 package manager program (**pip**) in **venv** started when invoking the **edrc/bin/pip** command.

EXAMPLES

-

SEE ALSO

edrcintro(1), **pip3(1)**, **pkgdir(1m)**, **pythonenv(3)**, **pythonversion(3)**, **python_wrapper(3)**,

NOTES

-

BUGS

-

AUTHOR

pip was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pip3 – package manager for Python packages

SYNOPSIS

pip3 <command> [options]

DESCRIPTION

pip3 is the PyPA recommended package manager for Python packages

OPTIONS

-h, --help

Show help.

(environment variable: **PIP_HELP**)

--debug

Let unhandled exceptions propagate outside the main subroutine, instead of logging them to stderr.

(environment variable: **PIP_DEBUG**)

--isolated

Run pip in an isolated mode, ignoring environment variables and user configuration.

(environment variable: **PIP_ISOLATED**)

--require-virtualenv

Allow pip to only run in a virtual environment; exit with an error otherwise.

(environment variable: **PIP_REQUIRE_VIRTUALENV**, **PIP_REQUIRE_VENV**)

--python <python>

Run pip with the specified Python interpreter.

(environment variable: **PIP_PYTHON**)

-v, --verbose

Give more output. Option is additive, and can be used up to 3 times.

(environment variable: **PIP_VERBOSE**)

-V, --version

Show version and exit.

(environment variable: **PIP_VERSION**)

-q, --quiet

Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).

(environment variable: **PIP_QUIET**)

--log <path>

Path to a verbose appending log.

(environment variable: **PIP_LOG**, **PIP_LOG_FILE**, **PIP_LOCAL_LOG**)

--no-input

Disable prompting for input.

(environment variable: **PIP_NO_INPUT**)

--keyring-provider <keyring_provider>

Enable the credential lookup via the keyring library if user input is allowed. Specify which mechanism to use [disabled, import, subprocess]. (default: disabled)

(environment variable: **PIP_KEYRING_PROVIDER**)

--proxy <proxy>

Specify a proxy in the form scheme://[user:passwd@]proxy.server:port.

(environment variable: **PIP_PROXY**)

--retries <retries>

Maximum number of retries each connection should attempt (default 5 times).

(environment variable: **PIP_RETRIES**)

--timeout <sec>

Set the socket timeout (default 15 seconds).

(environment variable: **PIP_TIMEOUT**, **PIP_DEFAULT_TIMEOUT**)

--exists-action <action>

Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.

(environment variable: **PIP_EXISTS_ACTION**)

--trusted-host <hostname>

Mark this host or host:port pair as trusted, even though it does not have valid or any HTTPS.

(environment variable: **PIP_TRUSTED_HOST**)

--cert <path>

Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL Certificate Verification' in pip documentation for more information.

(environment variable: **PIP_CERT**)

--client-cert <path>

Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.

(environment variable: **PIP_CLIENT_CERT**)

--cache-dir <dir>

Store the cache data in <dir>.

(environment variable: **PIP_CACHE_DIR**)

--no-cache-dir

Disable the cache.

(environment variable: **PIP_NO_CACHE_DIR**)

--disable-pip-version-check

Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with **--no-index**.

(environment variable: **PIP_DISABLE_PIP_VERSION_CHECK**)

--no-color

Suppress colored output.

(environment variable: **PIP_NO_COLOR**)

--no-python-version-warning

Silence deprecation warnings for upcoming unsupported Pythons.

(environment variable: **PIP_NO_PYTHON_VERSION_WARNING**)

--use-feature <feature>

Enable new functionality, that may be backward incompatible.

(environment variable: **PIP_USE_FEATURE**)

--use-deprecated <feature>

Enable deprecated functionality, that will be removed in the future.

(environment variable: **PIP_USE_DEPRECATED**)

COMMANDS**pip3-install(1)**

Install packages.

pip3-download(1)

Download packages.

pip3-uninstall(1)

Uninstall packages.

pip3-freeze(1)

Output installed packages in requirements format.

pip3-list(1)

List installed packages.

pip3-show(1)

Show information about installed packages.

pip3-check(1)

Verify installed packages have compatible dependencies.

pip3-search(1)

Search PyPI for packages.

pip3-wheel(1)

Build wheels from your requirements.

pip3-hash(1)

Compute hashes of package archives.

pip3-help(1)

Show help for pip commands.

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-check – description of pip3 check command

DESCRIPTION

Verify installed packages have compatible dependencies.

USAGE

```
python -m pip check [options]
```

OPTIONS

-h, --help

Show help.

(environment variable: **PIP_HELP**)

--debug

Let unhandled exceptions propagate outside the main subroutine, instead of logging them to stderr.

(environment variable: **PIP_DEBUG**)

--isolated

Run pip in an isolated mode, ignoring environment variables and user configuration.

(environment variable: **PIP_ISOLATED**)

--require-virtualenv

Allow pip to only run in a virtual environment; exit with an error otherwise.

(environment variable: **PIP_REQUIRE_VIRTUALENV**, **PIP_REQUIRE_VENV**)

--python <python>

Run pip with the specified Python interpreter.

(environment variable: **PIP_PYTHON**)

-v, --verbose

Give more output. Option is additive, and can be used up to 3 times.

(environment variable: **PIP_VERBOSE**)

-V, --version

Show version and exit.

(environment variable: **PIP_VERSION**)

-q, --quiet

Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).

(environment variable: **PIP_QUIET**)

--log <path>

Path to a verbose appending log.

(environment variable: **PIP_LOG**, **PIP_LOG_FILE**, **PIP_LOCAL_LOG**)

--no-input

Disable prompting for input.

(environment variable: **PIP_NO_INPUT**)

--keyring-provider <keyring_provider>

Enable the credential lookup via the keyring library if user input is allowed. Specify which mechanism to use [disabled, import, subprocess]. (default: disabled)

(environment variable: **PIP_KEYRING_PROVIDER**)

--proxy <proxy>

Specify a proxy in the form scheme://[user:passwd@]proxy.server:port.

(environment variable: **PIP_PROXY**)

--retries <retries>

Maximum number of retries each connection should attempt (default 5 times).

(environment variable: **PIP_RETRIES**)

--timeout <sec>

Set the socket timeout (default 15 seconds).

(environment variable: **PIP_TIMEOUT**, **PIP_DEFAULT_TIMEOUT**)

--exists-action <action>

Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.

(environment variable: **PIP_EXISTS_ACTION**)

--trusted-host <hostname>

Mark this host or host:port pair as trusted, even though it does not have valid or any HTTPS.

(environment variable: **PIP_TRUSTED_HOST**)

--cert <path>

Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL Certificate Verification' in pip documentation for more information.

(environment variable: **PIP_CERT**)

--client-cert <path>

Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.

(environment variable: **PIP_CLIENT_CERT**)

--cache-dir <dir>

Store the cache data in <dir>.

(environment variable: **PIP_CACHE_DIR**)

--no-cache-dir

Disable the cache.

(environment variable: **PIP_NO_CACHE_DIR**)

--disable-pip-version-check

Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with **--no-index**.

(environment variable: **PIP_DISABLE_PIP_VERSION_CHECK**)

--no-color

Suppress colored output.

(environment variable: **PIP_NO_COLOR**)

--no-python-version-warning

Silence deprecation warnings for upcoming unsupported Pythons.

(environment variable: **PIP_NO_PYTHON_VERSION_WARNING**)

--use-feature <feature>

Enable new functionality, that may be backward incompatible.

(environment variable: **PIP_USE_FEATURE**)

--use-deprecated <feature>

Enable deprecated functionality, that will be removed in the future.

(environment variable: **PIP_USE_DEPRECATED**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-download – description of pip3 download command

DESCRIPTION

Download packages from:

- PyPI (and other indexes) using requirement specifiers.
- VCS project urls.
- Local project directories.
- Local or remote source archives.

pip also supports downloading from "requirements files", which provide an easy way to specify a whole environment to be downloaded.

USAGE

```
python -m pip download [options] <requirement specifier> [package-index-options]
python -m pip download [options] -r <requirements file> [package-index-options] .
python -m pip download [options] <vcs project url> ...
python -m pip download [options] <local project path> ...
python -m pip download [options] <archive url/path> ...
```

OPTIONS

-c, --constraint <file>

Constrain versions using the given constraints file. This option can be used multiple times.

(environment variable: **PIP_CONSTRAINT**)

-r, --requirement <file>

Install from the given requirements file. This option can be used multiple times.

(environment variable: **PIP_REQUIREMENT**)

--no-deps

Don't install package dependencies.

(environment variable: **PIP_NO_DEPS, PIP_NO_DEPENDENCIES**)

--global-option <options>

Extra global options to be supplied to the setup.py call before the install or bdist_wheel command.

(environment variable: **PIP_GLOBAL_OPTION**)

--no-binary <format_control>

Do not use binary packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all binary packages, ":none:" to empty the set (notice the colons), or one or more package names with commas between them (no colons). Note that some packages are tricky to compile and may fail to install when this option is used on them.

(environment variable: **PIP_NO_BINARY**)

--only-binary <format_control>

Do not use source packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all source packages, ":none:" to empty the set, or one or more package names with commas between them. Packages without binary distributions will fail to install when this option is used on them.

(environment variable: **PIP_ONLY_BINARY**)

--prefer-binary

Prefer binary packages over source packages, even if the source packages are newer.

(environment variable: **PIP_PREFER_BINARY**)

--src <dir>

Directory to check out editable projects into. The default in a virtualenv is "<venv path>/src". The default for global installs is "<current dir>/src".

(environment variable: **PIP_SRC**, **PIP_SOURCE**, **PIP_SOURCE_DIR**, **PIP_SOURCE_DIRECTORY**)

--pre Include pre-release and development versions. By default, pip only finds stable versions.

(environment variable: **PIP_PRE**)

--require-hashes

Require a hash to check each requirement against, for repeatable installs. This option is implied when any package in a requirements file has a **--hash** option.

(environment variable: **PIP_REQUIRE_HASHES**)

--progress-bar <progress_bar>

Specify whether the progress bar should be used [on, off, raw] (default: on)

(environment variable: **PIP_PROGRESS_BAR**)

--no-build-isolation

Disable isolation when building a modern source distribution. Build dependencies specified by PEP 518 must be already installed if this option is used.

(environment variable: **PIP_NO_BUILD_ISOLATION**)

--use-pep517

Use PEP 517 for building source distributions (use **--no-use-pep517** to force legacy behaviour).

(environment variable: **PIP_USE_PEP517**)

--check-build-dependencies

Check the build dependencies when PEP517 is used.

(environment variable: **PIP_CHECK_BUILD_DEPENDENCIES**)

--ignore-requires-python

Ignore the Requires-Python information.

(environment variable: **PIP_IGNORE_REQUIRES_PYTHON**)

-d, --dest <dir>

Download packages into <dir>.

(environment variable: **PIP_DEST**, **PIP_DESTINATION_DIR**, **PIP_DESTINATION_DIRECTORY**)

--platform <platform>

Only use wheels compatible with <platform>. Defaults to the platform of the running system. Use this option multiple times to specify multiple platforms supported by the target interpreter.

(environment variable: **PIP_PLATFORM**)

--python-version <python_version>

The Python interpreter version to use for wheel and "Requires-Python" compatibility checks. Defaults to a version derived from the running interpreter. The version can be specified using up to three dot-separated integers (e.g. "3" for 3.0.0, "3.7" for 3.7.0, or "3.7.3"). A major-minor version can also be given as a string without dots (e.g. "37" for 3.7.0).

(environment variable: **PIP_PYTHON_VERSION**)

--implementation <implementation>

Only use wheels compatible with Python implementation <implementation>, e.g. 'pp', 'jy', 'cp', or 'ip'. If not specified, then the current interpreter implementation is used. Use 'py' to force implementation-agnostic wheels.

(environment variable: **PIP_IMPLEMENTATION**)

--abi <abi>

Only use wheels compatible with Python abi <abi>, e.g. 'pypy_41'. If not specified, then the current interpreter abi tag is used. Use this option multiple times to specify multiple abis supported by the target interpreter. Generally you will need to specify --implementation, --platform, and --python-version when using this option.

(environment variable: **PIP_ABI**)

--no-clean

Don't clean up build directories.

(environment variable: **PIP_NO_CLEAN**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-freeze – description of pip3 freeze command

DESCRIPTION

Output installed packages in requirements format.

packages are listed in a case-insensitive sorted order.

USAGE

```
python -m pip freeze [options]
```

OPTIONS

-r, --requirement <file>

Use the order in the given requirements file and its comments when generating output. This option can be used multiple times.

(environment variable: **PIP_REQUIREMENT**)

-l, --local

If in a virtualenv that has global access, do not output globally-installed packages.

(environment variable: **PIP_LOCAL**)

--user Only output packages installed in user-site.

(environment variable: **PIP_USER**)

--path <path>

Restrict to the specified installation path for listing packages (can be used multiple times).

(environment variable: **PIP_PATH**)

--all Do not skip these packages in the output: pip

(environment variable: **PIP_ALL**)

--exclude-editable

Exclude editable package from output.

(environment variable: **PIP_EXCLUDE_EDITABLE**)

--exclude <package>

Exclude specified package from the output

(environment variable: **PIP_EXCLUDE**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-hash – description of pip3 hash command

DESCRIPTION

Compute a hash of a local package archive.

These can be used with `--hash` in a requirements file to do repeatable installs.

USAGE

```
python -m pip hash [options] <file> ...
```

OPTIONS

-a, --algorithm <algorithm>

The hash algorithm to use: one of sha256, sha384, sha512

(environment variable: **PIP_ALGORITHM**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-help – description of pip3 help command

DESCRIPTION

Show help for commands

USAGE

python -m pip help <command>

OPTIONS

-h, --help

Show help.

(environment variable: **PIP_HELP**)

--debug

Let unhandled exceptions propagate outside the main subroutine, instead of logging them to stderr.

(environment variable: **PIP_DEBUG**)

--isolated

Run pip in an isolated mode, ignoring environment variables and user configuration.

(environment variable: **PIP_ISOLATED**)

--require-virtualenv

Allow pip to only run in a virtual environment; exit with an error otherwise.

(environment variable: **PIP_REQUIRE_VIRTUALENV**, **PIP_REQUIRE_VENV**)

--python <python>

Run pip with the specified Python interpreter.

(environment variable: **PIP_PYTHON**)

-v, --verbose

Give more output. Option is additive, and can be used up to 3 times.

(environment variable: **PIP_VERBOSE**)

-V, --version

Show version and exit.

(environment variable: **PIP_VERSION**)

-q, --quiet

Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).

(environment variable: **PIP_QUIET**)

--log <path>

Path to a verbose appending log.

(environment variable: **PIP_LOG**, **PIP_LOG_FILE**, **PIP_LOCAL_LOG**)

--no-input

Disable prompting for input.

(environment variable: **PIP_NO_INPUT**)

--keyring-provider <keyring_provider>

Enable the credential lookup via the keyring library if user input is allowed. Specify which mechanism to use [disabled, import, subprocess]. (default: disabled)

(environment variable: **PIP_KEYRING_PROVIDER**)

--proxy <proxy>

Specify a proxy in the form scheme://[user:passwd@]proxy.server:port.

(environment variable: **PIP_PROXY**)

--retries <retries>

Maximum number of retries each connection should attempt (default 5 times).

(environment variable: **PIP_RETRIES**)

--timeout <sec>

Set the socket timeout (default 15 seconds).

(environment variable: **PIP_TIMEOUT**, **PIP_DEFAULT_TIMEOUT**)

--exists-action <action>

Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.

(environment variable: **PIP_EXISTS_ACTION**)

--trusted-host <hostname>

Mark this host or host:port pair as trusted, even though it does not have valid or any HTTPS.

(environment variable: **PIP_TRUSTED_HOST**)

--cert <path>

Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL Certificate Verification' in pip documentation for more information.

(environment variable: **PIP_CERT**)

--client-cert <path>

Path to SSL client certificate, a single file containing the private key and the certificate in PEM format.

(environment variable: **PIP_CLIENT_CERT**)

--cache-dir <dir>

Store the cache data in <dir>.

(environment variable: **PIP_CACHE_DIR**)

--no-cache-dir

Disable the cache.

(environment variable: **PIP_NO_CACHE_DIR**)

--disable-pip-version-check

Don't periodically check PyPI to determine whether a new version of pip is available for download. Implied with **--no-index**.

(environment variable: **PIP_DISABLE_PIP_VERSION_CHECK**)

--no-color

Suppress colored output.

(environment variable: **PIP_NO_COLOR**)

--no-python-version-warning

Silence deprecation warnings for upcoming unsupported Pythons.

(environment variable: **PIP_NO_PYTHON_VERSION_WARNING**)

--use-feature <feature>

Enable new functionality, that may be backward incompatible.

(environment variable: **PIP_USE_FEATURE**)

--use-deprecated <feature>

Enable deprecated functionality, that will be removed in the future.

(environment variable: **PIP_USE_DEPRECATED**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-install – description of pip3 install command

DESCRIPTION

Install packages from:

- PyPI (and other indexes) using requirement specifiers.
- VCS project urls.
- Local project directories.
- Local or remote source archives.

pip also supports installing from "requirements files", which provide an easy way to specify a whole environment to be installed.

USAGE

```
python -m pip install [options] <requirement specifier> [package-index-options] .
python -m pip install [options] -r <requirements file> [package-index-options] ..
python -m pip install [options] [-e] <vcs project url> ...
python -m pip install [options] [-e] <local project path> ...
python -m pip install [options] <archive url/path> ...
```

OPTIONS

-r, --requirement <file>

Install from the given requirements file. This option can be used multiple times.

(environment variable: **PIP_REQUIREMENT**)

-c, --constraint <file>

Constrain versions using the given constraints file. This option can be used multiple times.

(environment variable: **PIP_CONSTRAINT**)

--no-deps

Don't install package dependencies.

(environment variable: **PIP_NO_DEPS, PIP_NO_DEPENDENCIES**)

--pre Include pre-release and development versions. By default, pip only finds stable versions.

(environment variable: **PIP_PRE**)

-e, --editable <path/url>

Install a project in editable mode (i.e. setuptools "develop mode") from a local project path or a VCS url.

(environment variable: **PIP_EDITABLE**)

--dry-run

Don't actually install anything, just print what would be. Can be used in combination with **--ignore-installed** to 'resolve' the requirements.

(environment variable: **PIP_DRY_RUN**)

-t, --target <dir>

Install packages into <dir>. By default this will not replace existing files/folders in <dir>. Use **--upgrade** to replace existing packages in <dir> with new versions.

(environment variable: **PIP_TARGET**)

--platform <platform>

Only use wheels compatible with <platform>. Defaults to the platform of the running system. Use this option multiple times to specify multiple platforms supported by the target interpreter.

(environment variable: **PIP_PLATFORM**)

--python-version <python_version>

The Python interpreter version to use for wheel and "Requires-Python" compatibility checks. Defaults to a version derived from the running interpreter. The version can be specified using up to three dot-separated integers (e.g. "3" for 3.0.0, "3.7" for 3.7.0, or "3.7.3"). A major-minor version can also be given as a string without dots (e.g. "37" for 3.7.0).

(environment variable: **PIP_PYTHON_VERSION**)

--implementation <implementation>

Only use wheels compatible with Python implementation <implementation>, e.g. 'pp', 'jy', 'cp', or 'ip'. If not specified, then the current interpreter implementation is used. Use 'py' to force implementation-agnostic wheels.

(environment variable: **PIP_IMPLEMENTATION**)

--abi <abi>

Only use wheels compatible with Python abi <abi>, e.g. 'pypy_41'. If not specified, then the current interpreter abi tag is used. Use this option multiple times to specify multiple abis supported by the target interpreter. Generally you will need to specify **--implementation**, **--platform**, and **--python-version** when using this option.

(environment variable: **PIP_ABI**)

--user Install to the Python user install directory for your platform. Typically ~/.local/, or %APP-DATA%\Python on Windows. (See the Python documentation for site.USER_BASE for full details.)

(environment variable: **PIP_USER**)

--root <dir>

Install everything relative to this alternate root directory.

(environment variable: **PIP_ROOT**)

--prefix <dir>

Installation prefix where lib, bin and other top-level folders are placed. Note that the resulting installation may contain scripts and other resources which reference the Python interpreter of pip, and not that of **--prefix**. See also the **--python** option if the intention is to install packages into another (possibly pip-free) environment.

(environment variable: **PIP_PREFIX**)

--src <dir>

Directory to check out editable projects into. The default in a virtualenv is "<venv path>/src". The default for global installs is "<current dir>/src".

(environment variable: **PIP_SRC**, **PIP_SOURCE**, **PIP_SOURCE_DIR**, **PIP_SOURCE_DIRECTORY**)

-U, --upgrade

Upgrade all specified packages to the newest available version. The handling of dependencies depends on the upgrade-strategy used.

(environment variable: **PIP_UPGRADE**)

--upgrade-strategy <upgrade_strategy>

Determines how dependency upgrading should be handled [default: only-if-needed]. "eager" – dependencies are upgraded regardless of whether the currently installed version satisfies the requirements of the upgraded package(s). "only-if-needed" – are upgraded only when they do not satisfy the requirements of the upgraded package(s).

(environment variable: **PIP_UPGRADE_STRATEGY**)

--force-reinstall

Reinstall all packages even if they are already up-to-date.

(environment variable: **PIP_FORCE_REINSTALL**)

-I, --ignore-installed

Ignore the installed packages, overwriting them. This can break your system if the existing package is of a different version or was installed with a different package manager!

(environment variable: **PIP_IGNORE_INSTALLED**)

--ignore-requires-python

Ignore the Requires-Python information.

(environment variable: **PIP_IGNORE_REQUIRES_PYTHON**)

--no-build-isolation

Disable isolation when building a modern source distribution. Build dependencies specified by PEP 518 must be already installed if this option is used.

(environment variable: **PIP_NO_BUILD_ISOLATION**)

--use-pep517

Use PEP 517 for building source distributions (use **--no-use-pep517** to force legacy behaviour).

(environment variable: **PIP_USE_PEP517**)

--check-build-dependencies

Check the build dependencies when PEP517 is used.

(environment variable: **PIP_CHECK_BUILD_DEPENDENCIES**)

--break-system-packages

Allow pip to modify an EXTERNALLY-MANAGED Python installation

(environment variable: **PIP_BREAK_SYSTEM_PACKAGES**)

-C, --config-settings <settings>

Configuration settings to be passed to the PEP 517 build backend. Settings take the form KEY=VALUE. Use multiple **--config-settings** options to pass multiple keys to the backend.

(environment variable: **PIP_CONFIG_SETTINGS**)

--global-option <options>

Extra global options to be supplied to the setup.py call before the install or bdist_wheel command.

(environment variable: **PIP_GLOBAL_OPTION**)

--compile

Compile Python source files to bytecode

(environment variable: **PIP_COMPILE**)

--no-compile

Do not compile Python source files to bytecode

(environment variable: **PIP_NO_COMPILE**)

--no-warn-script-location

Do not warn when installing scripts outside PATH

(environment variable: **PIP_NO_WARN_SCRIPT_LOCATION**)

--no-warn-conflicts

Do not warn about broken dependencies

(environment variable: **PIP_NO_WARN_CONFLICTS**)

--no-binary <format_control>

Do not use binary packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all binary packages, ":none:" to empty the set (notice the colons), or one or more package names with commas between them (no colons). Note that some packages are tricky to compile and may fail to install when this option is used on them.

(environment variable: **PIP_NO_BINARY**)

--only-binary <format_control>

Do not use source packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all source packages, ":none:" to empty the set, or one or more package names with commas between them. Packages without binary distributions will fail to install when this option is used on them.

(environment variable: **PIP_ONLY_BINARY**)

--prefer-binary

Prefer binary packages over source packages, even if the source packages are newer.

(environment variable: **PIP_PREFER_BINARY**)

--require-hashes

Require a hash to check each requirement against, for repeatable installs. This option is implied when any package in a requirements file has a `---hash` option.

(environment variable: **PIP_REQUIRE_HASHES**)

--progress-bar <progress_bar>

Specify whether the progress bar should be used [on, off, raw] (default: on)

(environment variable: **PIP_PROGRESS_BAR**)

--root-user-action <root_user_action>

Action if pip is run as a root user [warn, ignore] (default: warn)

(environment variable: **PIP_ROOT_USER_ACTION**)

--report <file>

Generate a JSON file describing what pip did to install the provided requirements. Can be used in combination with `--dry-run` and `--ignore-installed` to 'resolve' the requirements. When `-` is used as file name it writes to stdout. When writing to stdout, please combine with the `--quiet` option to avoid mixing pip logging output with JSON output.

(environment variable: **PIP_REPORT**)

--no-clean

Don't clean up build directories.

(environment variable: **PIP_NO_CLEAN**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-list – description of pip3 list command

DESCRIPTION

List installed packages, including editables.

Packages are listed in a case-insensitive sorted order.

USAGE

```
python -m pip list [options]
```

OPTIONS**-o, --outdated**

List outdated packages

(environment variable: **PIP_OUTDATED**)

-u, --uptodate

List uptodate packages

(environment variable: **PIP_UPTODATE**)

-e, --editable

List editable projects.

(environment variable: **PIP_EDITABLE**)

-l, --local

If in a virtualenv that has global access, do not list globally-installed packages.

(environment variable: **PIP_LOCAL**)

--user Only output packages installed in user-site.

(environment variable: **PIP_USER**)

--path <path>

Restrict to the specified installation path for listing packages (can be used multiple times).

(environment variable: **PIP_PATH**)

--pre Include pre-release and development versions. By default, pip only finds stable versions.

(environment variable: **PIP_PRE**)

--format <list_format>

Select the output format among: columns (default), freeze, or json. The 'freeze' format cannot be used with the **--outdated** option.

(environment variable: **PIP_FORMAT**)

--not-required

List packages that are not dependencies of installed packages.

(environment variable: **PIP_NOT_REQUIRED**)

--exclude-editable

Exclude editable package from output.

(environment variable: **PIP_EXCLUDE_EDITABLE**)

--include-editable

Include editable package from output.

(environment variable: **PIP_INCLUDE_EDITABLE**)

--exclude <package>

Exclude specified package from the output

(environment variable: **PIP_EXCLUDE**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-search – description of pip3 search command

DESCRIPTION

Search for PyPI packages whose name or summary contains <query>.

USAGE

python -m pip search [options] <query>

OPTIONS

-i, --index <url>

Base URL of Python Package Index (default *https://pypi.org/pypi*)

(environment variable: **PIP_INDEX**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-show – description of pip3 show command

DESCRIPTION

Show information about one or more installed packages.

The output is in RFC-compliant mail header format.

USAGE

```
python -m pip show [options] <package> ...
```

OPTIONS

-f, --files

Show the full list of installed files for each package.

(environment variable: **PIP_FILES**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-uninstall – description of pip3 uninstall command

DESCRIPTION

Uninstall packages.

pip is able to uninstall most installed packages. Known exceptions are:

- Pure distutils packages installed with **python setup.py install**, which leave behind no metadata to determine what files were installed.
- Script wrappers installed by **python setup.py develop**.

USAGE

```
python -m pip uninstall [options] <package> ...
python -m pip uninstall [options] -r <requirements file> ...
```

OPTIONS

-r, --requirement <file>

Uninstall all the packages listed in the given requirements file. This option can be used multiple times.

(environment variable: **PIP_REQUIREMENT**)

-y, --yes

Don't ask for confirmation of uninstall deletions.

(environment variable: **PIP_YES**)

--root-user-action <root_user_action>

Action if pip is run as a root user [warn, ignore] (default: warn)

(environment variable: **PIP_ROOT_USER_ACTION**)

--break-system-packages

Allow pip to modify an EXTERNALLY-MANAGED Python installation

(environment variable: **PIP_BREAK_SYSTEM_PACKAGES**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pip3-wheel – description of pip3 wheel command

DESCRIPTION

Build Wheel archives for your requirements and dependencies.

Wheel is a built-package format, and offers the advantage of not recompiling your software during every install. For more details, see the wheel docs: <https://wheel.readthedocs.io/en/latest/>

'pip wheel' uses the build system interface as described here: - <https://pip.pypa.io/en/stable/reference/build-system/>

USAGE

```
python -m pip wheel [options] <requirement specifier> ...
python -m pip wheel [options] -r <requirements file> ...
python -m pip wheel [options] [-e] <vcs project url> ...
python -m pip wheel [options] [-e] <local project path> ...
python -m pip wheel [options] <archive url/path> ...
```

OPTIONS

-w, --wheel-dir <dir>

Build wheels into <dir>, where the default is the current working directory.

(environment variable: **PIP_WHEEL_DIR**)

--no-binary <format_control>

Do not use binary packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all binary packages, ":none:" to empty the set (notice the colons), or one or more package names with commas between them (no colons). Note that some packages are tricky to compile and may fail to install when this option is used on them.

(environment variable: **PIP_NO_BINARY**)

--only-binary <format_control>

Do not use source packages. Can be supplied multiple times, and each time adds to the existing value. Accepts either ":all:" to disable all source packages, ":none:" to empty the set, or one or more package names with commas between them. Packages without binary distributions will fail to install when this option is used on them.

(environment variable: **PIP_ONLY_BINARY**)

--prefer-binary

Prefer binary packages over source packages, even if the source packages are newer.

(environment variable: **PIP_PREFER_BINARY**)

--no-build-isolation

Disable isolation when building a modern source distribution. Build dependencies specified by PEP 518 must be already installed if this option is used.

(environment variable: **PIP_NO_BUILD_ISOLATION**)

--use-pep517

Use PEP 517 for building source distributions (use **--no-use-pep517** to force legacy behaviour).

(environment variable: **PIP_USE_PEP517**)

--check-build-dependencies

Check the build dependencies when PEP517 is used.

(environment variable: **PIP_CHECK_BUILD_DEPENDENCIES**)

-c, --constraint <file>

Constrain versions using the given constraints file. This option can be used multiple times.

(environment variable: **PIP_CONSTRAINT**)

-e, --editable <path/url>

Install a project in editable mode (i.e. setuptools "develop mode") from a local project path or a VCS url.

(environment variable: **PIP_EDITABLE**)

-r, --requirement <file>

Install from the given requirements file. This option can be used multiple times.

(environment variable: **PIP_REQUIREMENT**)

--src <dir>

Directory to check out editable projects into. The default in a virtualenv is "<venv path>/src". The default for global installs is "<current dir>/src".

(environment variable: **PIP_SRC, PIP_SOURCE, PIP_SOURCE_DIR, PIP_SOURCE_DIRECTORY**)

--ignore-requires-python

Ignore the Requires-Python information.

(environment variable: **PIP_IGNORE_REQUIRES_PYTHON**)

--no-deps

Don't install package dependencies.

(environment variable: **PIP_NO_DEPS, PIP_NO_DEPENDENCIES**)

--progress-bar <progress_bar>

Specify whether the progress bar should be used [on, off, raw] (default: on)

(environment variable: **PIP_PROGRESS_BAR**)

--no-verify

Don't verify if built wheel is valid.

(environment variable: **PIP_NO_VERIFY**)

-C, --config-settings <settings>

Configuration settings to be passed to the PEP 517 build backend. Settings take the form KEY=VALUE. Use multiple --config-settings options to pass multiple keys to the backend.

(environment variable: **PIP_CONFIG_SETTINGS**)

--build-option <options>

Extra arguments to be supplied to 'setup.py bdist_wheel'.

(environment variable: **PIP_BUILD_OPTION**)

--global-option <options>

Extra global options to be supplied to the setup.py call before the install or bdist_wheel command.

(environment variable: **PIP_GLOBAL_OPTION**)

--pre Include pre-release and development versions. By default, pip only finds stable versions.

(environment variable: **PIP_PRE**)

--require-hashes

Require a hash to check each requirement against, for repeatable installs. This option is implied when any package in a requirements file has a **--hash** option.

(environment variable: **PIP_REQUIRE_HASHES**)

--no-clean

Don't clean up build directories.

(environment variable: **PIP_NO_CLEAN**)

AUTHOR

pip developers

COPYRIGHT

The pip developers

NAME

pkgdir – directory of package/software handling commands on different operating systems

SYNOPSIS

-

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is a directory of operating system package handling commands for different operating systems.

OVERVIEW

| GRP | | OSID | DESCRIPTION |
|-----|--|-------------------------------|----------------------|
| RPM | | Linux Linux-64 Linux-ia | RedHat Linux based |
| DEB | | Linux Linux-64 | Debian Linux based |
| HPX | | HP-10 HP-11 HP-11ia | Hewlett Packard Unix |
| SOL | | Solaris Solaris-x86 | Sun System-5 Unix |
| SUN | | SunOS | Sun BSD Unix |
| AIX | | AIX | IBM Unix |
| OSF | | DEC-OSF1 | Digital Unix |

| GRP | | OSID | DESCRIPTION |
|-----|--|------|-------------------------------------|
| PIP | | * | Python Package Installation (pip) |
| ROK | | * | Lua Package Installation (luarocks) |
| CPA | | * | Perl Package Installation (cpanm) |

RPM - RedHat (rpm) Linux based**ACT:** QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config**SRC:** S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|-------------|--|
| QRY | S | all | rpm --query --all rpm -qa |
| | S | detail info | rpm --query --info <i>coreutils</i> rpm -qi <i>coreutils</i> |
| | S | changelog | rpm --query --changelog <i>coreutils</i> rpm -q --changelog <i>coreutils</i> |
| | S | files | rpm --query --list <i>coreutils</i> rpm -ql <i>coreutils</i> |
| | S | pkg of file | rpm --query --whatprovides <i>/bin/ls</i> rpm -q --whatprovides <i>/bin/ls</i> |
| | S | details | rpm --query --all --queryformat \ " %{NAME}-%{VERSION}-%{RELEASE} %{INSTALLTIME:date} %{SUMMARY}n" rpm -qa --qf \ " %{NAME}-%{VERSION}-%{RELEASE} %{INSTALLTIME:date} %{SUMMARY}n" |
| | S | scripts | rpm --query --scripts <i>coreutils</i> rpm -q --scripts <i>coreutils</i> |
| | F | detail info | rpm --query --package --info <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -qpi <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | files | rpm --query --package --list <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -qpl <i>man-2.6.3-4.1.3.x86_64.rpm</i> <i>lscomp man-2.6.3-4.1.3.x86_64.rpm</i> <i>llcomp man-2.6.3-4.1.3.x86_64.rpm</i> <i>catcomp man-2.6.3-4.1.3.x86_64.rpm /etc/manpath.config</i> <i>unrpm man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | changelog | rpm --query --changelog <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -q --changelog <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | scripts | rpm --query --package --scripts <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -qp --scripts <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | | | |
| | | | |
| | | | |

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|--------------------------|--|
| INS | F | install | rpm --install --verbose <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -iv <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | no depend- ency check | rpm --install --verbose --nodeps <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -iv --nodeps <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | force install | rpm --install --verbose --force <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -iv --force <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| UPD | F | update | rpm --update --verbose <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -Uv <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | no depend- ency check | rpm --update --verbose --nodeps <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -Uv --nodeps <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| | F | force update | rpm --update --verbose --force <i>man-2.6.3-4.1.3.x86_64.rpm</i> rpm -Uv --force <i>man-2.6.3-4.1.3.x86_64.rpm</i> |
| DEL | S | uninstall | rpm --erase --verbose <i>man</i> rpm -ev <i>man</i> |
| BLD | | | |
| CFG | | | |

See: **rpm(8)**, **rpmbuild(8)**.

On systems lacking the rpm packages query the CentOS man pages using **edrcman -o CentOS manpage**.

DEB - Debian Linux based

ACT: QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|-------------|--|
| QRY | S | all | dpkg-query --list dpkg-query -l |
| | S | all | dpkg-query --show dpkg-query -W |
| | S | pkg search | dpkg-query --list 'opera*' dpkg-query -l 'opera*' |
| | S | pkg search | dpkg-query --show 'opera*' dpkg-query -W 'opera*' |
| | S | detail info | dpkg-query --status coreutils dpkg-query -s coreutils |
| | S | files | dpkg-query --listfiles coreutils dpkg-query -L coreutils |
| | S | pkg of file | dpkg-query --search /bin/ls dpkg-query -S /bin/ls |
| | S | details | dpkg-query --show \ --showformat='\${binary:Package}-\${Version} \${binary:Summary}\n' dpkg-query -W \ -f='\${binary:Package}-\${Version} \${binary:Summary}\n' |
| | F | detail info | dpkg --info apt_1.6.17_amd64.deb dpkg -I apt_1.6.17_amd64.deb |
| | F | files | dpkg --contents apt_1.6.17_amd64.deb dpkg -c apt_1.6.17_amd64.deb lscomp apt_1.6.17_amd64.deb llcomp apt_1.6.17_amd64.deb catcomp apt_1.6.17_amd64.deb /lib/systemd/system/apt-daily.service undeb apt_1.6.17_amd64.deb |
| | R | detail info | apt-cache cache show coreutils |

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|--------------|---|
| INS | R | install | <code>apt-get install libslang2-dev</code> |
| | F | install | <code>dpkg --install opera-stable_90.0.4480.48_amd64.deb</code> <code>dpkg -i opera-stable_90.0.4480.48_amd64.deb</code> |
| UPD | R | update | <code>apt-get update && apt-get --only-upgrade install firefox</code> |
| | F | update | <code>dpkg --install opera-stable_90.0.4480.48_amd64.deb</code> <code>dpkg -i opera-stable_90.0.4480.48_amd64.deb</code> |
| DEL | S | uninstall | <code>apt remove libslang2-dev</code> |
| | S | uninstall | <code>dpkg --remove opera-stable</code> <code>dpkg -r opera-stable</code> |
| | S | purge config | <code>dpkg --purge opera-stable</code> <code>dpkg -P opera-stable</code> |
| BLD | | | |
| CFG | R | show config | <code>cat /etc/apt/sources.list</code> |
| | R | search | <code>apt-cache search coreutils</code> <code>apt-cache search .</code> |

See: `apt(8)`, `apt-cache(1)`, `dpkg(1)`, `dpkg-query(1)`, `apt-get(8)`.

On systems lacking the `apt/dpkg` packages query the Ubuntu man pages using **edrcman -o Ubuntu man-page**.

HPX - Hewlett Packard Unix

ACT: QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|---------|---------|
| QRY | | | |
| INS | | | |
| UPD | | | |
| DEL | | | |
| BLD | | | |
| CFG | | | |

See: **swdepot(1m)**.

On systems lacking the swdepot packages see **mandir(4)** to find online man page resources or use **edrc-man -i hpux manpage**.

SOL - Solaris

ACT: QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|-------------|----------------------------|
| QRY | S | detail info | pkginfo SUNWman |
| | F | detail info | pkginfo -d /dat/sw SUNWman |
| INS | F | install | pkgadd -d /dat/sw SUNWman |
| UPD | | | |
| DEL | S | uninstall | pkgrm SUNWman |
| BLD | | | |
| CFG | | | |

See: **pkg(1)**, **pkgadd(8)**, **pkginfo(1)**, **pkgmk(1)**, **pkgrm(8)**.

On systems lacking the pkg packages query the OpenSolaris man pages using **edrcman -o OpenSolaris manpage**.

SUN - SunOS

ACT: QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|---------|---------|
| QRY | | | |
| INS | | | |
| UPD | | | |
| DEL | | | |
| BLD | | | |
| CFG | | | |

See: -

On systems lacking the SunOS packages see **mandir**(4) to find online man page resources or use **edrcman -i sunos manpage**.

AIX - IBM Unix

ACT: QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|-----------|---|
| QRY | F | info | <code>installp -l -d NSH86-AIX32.bff</code> |
| INS | F | install | <code>installp -ac -d NSH86-AIX32.bff bladelogic.nsh</code> |
| UPD | F | update | <code>installp -ac -d NSH86-AIX32.bff bladelogic.nsh</code> |
| DEL | S | uninstall | <code>installp -u bladelogic.nsh</code> |
| BLD | | | |
| CFG | | | |

See: -

On systems lacking the installp packages see **mandir**(4) to find online man page resources.

OSF - Digital Unix

ACT: QRY=Query, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config
SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|---------|---------|
| QRY | | | |
| INS | | | |
| UPD | | | |
| DEL | | | |
| BLD | | | |
| CFG | | | |

See: -

On systems lacking the Digital Unix packages see **mandir(4)** to find online man page resources.

PIP - Python Package Installation

ACT: QRY=Query, DNL=Download, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|--------------------------|---|
| QRY | S | all | pip list |
| | S | WA2L/edrc venv | pip list --local |
| | S | detail info | pip show <i>complex</i> |
| | S | files | pip show --files <i>complex</i> pip show -f <i>complex</i> |
| | S | requirements | pip freeze --local > <i>requirements.txt</i> |
| | F | files | lscomp <i>complex-0.2.0.tar.gz</i> llcomp <i>complex-0.2.0.tar.gz</i> lscomp <i>complex-0.2.0-py3-none-any.whl</i> llcomp <i>complex-0.2.0-py3-none-any.whl</i> |
| | F | detail info | catcomp <i>complex-0.2.0.tar.gz \</i> <i>complex-0.2.0.tar.gz/PKG-INFO</i> catcomp <i>complex-0.2.0-py3-none-any.whl \</i> <i>complex-0.2.0.dist-info/METADATA</i> |
| DNL | R | source tar.gz | pip download --no-binary :all: <i>complex</i> |
| | R | wheel .whl | pip download <i>complex</i> |
| INS | R | install | pip install <i>complex</i> |
| | R | no depend- ency check | pip install --no-deps <i>complex</i> |
| | R | requirements | pip install -r <i>requirements.txt</i> |
| | F | install | pip install <i>complex-0.2.0.tar.gz</i> pip install <i>complex-0.2.0-py3-none-any.whl</i> |
| | F | no depend- ency check | pip install --no-deps <i>complex-0.2.0.tar.gz</i> pip install --no-deps <i>complex-0.2.0-py3-none-any.whl</i> |

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|-------------------|---|
| UPD | S | upgrade | <pre> pip install --upgrade <i>complex</i> pip install -U <i>complex</i> </pre> |
| | F | upgrade | <pre> pip install --upgrade <i>complex-0.2.0.tar.gz</i> pip install -U <i>complex-0.2.0.tar.gz</i> pip install --upgrade <i>complex-0.2.0-py3-none-any.whl</i> pip install -U <i>complex-0.2.0-py3-none-any.whl</i> </pre> |
| DEL | S | uninstall | <pre> pip uninstall <i>complex</i> </pre> |
| BLD | | | |
| CFG | S | venv | <pre> python3 -m venv --system-site-packages <i>/home/jdoe/py-venv</i> </pre> |
| | S | venv | <pre> python3 -m venv <i>/home/jdoe/py-venv</i> </pre> |
| | S | WA2L/edrc venv | <pre> edrc/lib/python/pym/.mk-venv </pre> |

See: **pip**(1), **pip3**(1), **pip3-check**(1), **pip3-download**(1), **pip3-freeze**(1), **pip3-hash**(1), **pip3-help**(1), **pip3-install**(1), **pip3-list**(1), **pip3-search**(1), **pip3-show**(1), **pip3-uninstall**(1), **pip3-wheel**(1), **pythonenv**(3)

The **edrc/bin/pip** command **pip**(1) works against the Python3 virtual environment (**venv**) distributed with WA2L/edrc located in the **edrc/lib/python/pym/<OSID>/<major>.<minor>/** directory.

ROK - Lua Package Installation

ACT: QRY=Query, DNL=Download, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|----------------------|--|
| QRY | S | all | luarocks list |
| | S | detail info | luarocks show <i>lcomplex</i> |
| | F | files | lscomp <i>lcomplex-20180729-1.src.ocks</i> llcomp <i>lcomplex-20180729-1.src.ocks</i> |
| | F | detail info | catcomp <i>lcomplex-20180729-1.src.ocks \</i> <i>lcomplex-20180729-1.ocks.spec</i> |
| | R | search | luarocks search <i>math</i> |
| DNL | R | .src.ocks | luarocks download <i>lcomplex</i> |
| INS | R | install | luarocks install <i>lcomplex</i> |
| | R | re-install | luarocks install <i>lcomplex --force</i> |
| | F | install | luarocks install <i>lcomplex-20180729-1.src.ocks</i> |
| UPD | R | update | luarocks install <i>lcomplex --force</i> |
| DEL | S | uninstall | luarocks uninstall <i>lcomplex</i> |
| BLD | | | |
| CFG | S | WA2L/edrc lua env | edrc/lib/lua/lum/.mk-struct |

See: [luaenv\(3\)](#), [luarocks\(1\)](#), [luarocks.cfg\(4\)](#), [luaenv\(3\)](#), [lua_wrapper\(1\)](#), <https://luarocks.org/>, <https://github.com/luarocks/luarocks/blob/main/docs/index.md>

The **edrc/bin/luarocks** command works against the Lua environment distributed with WA2L/edrc located in the **edrc/lib/lua/lum/<OSID>/<major>.<minor>/** directory.

After installing a package/module that provides compiled libraries into **edrc/lib/lua/lum/<OSID>/<major>.<minor>/lib/lua/<major>.<minor>/<GLIBC>/** ensure to manually create the related symbolic links to provide maximal portability.

CPA - Perl Package Installation

ACT: QRY=Query, DNL=Download, INS=Install, UPD=Update, DEL=Uninstall, BLD=Build, CFG=Config

SRC: S=System, F=File, R=Repository

| ACT | SRC | COMMENT | COMMAND |
|-----|-----|-------------|--|
| QRY | S | all | pmdesc |
| | S | all | lsrpm |
| | F | files | lscomp <i>MIME-Base64-3.16.tar.gz</i> llcomp <i>MIME-Base64-3.16.tar.gz</i> |
| | F | detail info | catcomp <i>MIME-Base64-3.16.tar.gz \</i> <i>MIME-Base64-3.16/META.json</i> |
| | F | detail info | catcomp <i>MIME-Base64-3.16.tar.gz \</i> <i>MIME-Base64-3.16/README</i> |
| DNL | R | .tar.gz | cpan -g <i>MIME::Base64</i> |
| INS | R | install | cpanm -i <i>MIME::Base64</i> |
| | F | install | cpanm -i <i>MIME-Base64-3.16.tar.gz</i> |
| UPD | R | update | cpanm -i <i>MIME::Base64</i> |
| | F | update | cpanm -i <i>MIME-Base64-3.16.tar.gz</i> |
| DEL | S | uninstall | cpanm -u <i>MIME::Base64</i> |
| BLD | | | |
| CFG | | | |

See: **cpanm**(1), **cpan**(1), **lsrpm**(1) **perlenv**(3), **perlversion**(3), **perl_wrapper**(1) **pmdesc**(1),

The **edrc/bin/cpanm** command works against the Perl environment distributed with WA2L/edrc located in the **edrc/lib/perl/pm/perl5/<major>.<minor>.<patch>/root/** directory.

SEE ALSO

edrcintro(1), **catcomp**(1), **edrcman**(1), **lscomp**(1), **llcomp**(1), **mandir**(4), **osid**(3), **undeb**(1), **unrpm**(1)

NOTES

-

BUGS

-

AUTHOR

pkgdir was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pkg_hostname – resolve package- or host name for the current dir

SYNOPSIS

edrc/lib/pkg_hostname

AVAILABILITY

WA2L/edrc

DESCRIPTION

return cluster package name the current working directory is related to.

If the directory is not related to a package the node name as resolved by **hostname**(1) is returned.

This command can be used to set the user's prompt depending on the current working directory of the user.

OPTIONS

-

ENVIRONMENT

\$OSID osid of the operating system as resolved by **osid**(3). If **\$OSID** is not set the OS-id is resolved by the **pkg_hostname** command.

EXIT STATUS

0 always.

FILES

etc/pkg_hostname.cfg

configuration file of **pkg_hostname**. In this file the mapping between the filesystems and the related cluster package is defined.

EXAMPLES

1) usage of **pkg_hostname** in the user .profile and .kshrc

Set the user prompt dynamically to show the user on which cluster the current working

directory is located.

~/.profile:

```

:
:

PKG_HOSTNAME=~edrc/lib/pkg_hostname`
SH=${SH:=`basename $SHELL`}

case "$SH" in
    *bash*)
        PS1="[ \${PWD} ]\n[ \${LOGNAME}@\\${PKG_HOSTNAME} ] [\$SH]: "
        ;;
    *)
        PS1="[ \${PWD} ]\n[ \${LOGNAME}@\\${PKG_HOSTNAME} ] [\$SH]: "
        ;;
esac

:
:
```

~/.kshrc:

```

:
:

ksh_cd(){
    if [ $# -eq 0 ]; then
        \cd $HOME
    else
        \cd "$*" 2>/dev/null
    fi
    ksh_cd_retval=$?
    if [ $ksh_cd_retval -eq 0 ]; then
        export PKG_HOSTNAME=~edrc/lib/pkg_hostname`
    else
        echo "ksh: $*: not found" >&2
        return $ksh_cd_retval
    fi
} # ksh_cd

alias cd='ksh_cd'

:
:
```

2) usage of pkg_hostname in a script

Suppose the script with the following cut-out is located in the path **/cluster/dwh_db1/app/bin** on the cluster packages in the TEST, PREPRODUCTION and PRODUCTION environment. The script has to run against different databases and has to mail a state information to different people depending on the package the script is running.

See also **server_environment(3)** for similar purpose.

```
#!/bin/sh

# Const
Scriptname=`basename $0`
Scriptpath=`dirname $0`

case `cd $Scriptpath; pkg_hostname` in
    dwh_dbl_tst)
        Mail_to=fred.flintstone@acme.ch
        Database=DWHTST
        ;;
    dwh_dbl_pre)
        Mail_to=barney.boulder@acme.ch
        Database=DWHPRE
        ;;
    dwh_dbl_prod)
        Mail_to=pete.pepple@acme.ch
        Database=DWHPROD
        ;;
    *)
        msg ERROR "check script!"
        exit 1
        ;;
esac

:
:
```

SEE ALSO

bash(1), **edrcintro(1)**, **getfilesystem(3)**, **ksh(1)**, **pkg_hostname.cfg(4)**, **server_environment(3)**

NOTES

-

BUGS

If the **getfilesystem(3)** command does not exist for a certain operating system, a slightly less precise mechanism is used to resolve on which filesystem the current working directory is located.

AUTHOR

pkg_hostname was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pkg_hostname.cfg – pkg_hostname configuration file

SYNOPSIS

etc/pkg_hostname.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the location definition for the **pkg_hostname** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

FILESYSTEM_REGEX;HOSTNAME_REGEX;PACKAGE_NAME;

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

FILESYSTEM_REGEX

regular expression to relate a filesystem (see output of the **getfilesystem(3)** command) to a cluster package name.

HOSTNAME_REGEX

regular expression to relate a filesystem on a certain host to a cluster package name.

PACKAGE_NAME

name of the cluster package. This name will be used to set the prompt when the current working directory is on a (moving) file system related to a cluster package.

EXAMPLES

```
#
# pkg_hostname.cfg - config file for pkg_hostname in PRODUCTION
#
# [00] 03.08.2004 CWa   Initial Version
#
```

```

#
# Format:
#
# <filesystem_regex>:<hostname_regex>:<package_name>:
#
/dev/vgadmb1/.*.*:adm_db1_prod:
/data_adm1/.*.*:adm_db1_prod:
adm_db1_prod.*.*:adm_db1_prod:

/dev/vgasyssv1/.*.*:asys_sv1_prod:
/data_asys1/.*.*:asys_sv1_prod:

/dev/vgbodb1/.*.*:bo_db1_prod:
/data_bo1/.*.*:bo_db1_prod:

/dev/vgbodb2/.*.*:bo_db2_prod:
/data_bo2/.*.*:bo_db2_prod:

/dev/vgbodb3/.*.*:bo_db3_prod:
/data_bo3/.*.*:bo_db3_prod:

/dev/vgbsi20/lvol4.*.*:ps_sv2_prod:
/data_ps/.*:dcdbsi20:ps_sv2_prod:

/dev/vgbsi21/lvol4.*.*:ps_sv3_prod:
/data_ps.*.*:dcdbsi21:ps_sv3_prod:

/dev/vgceddb1/.*.*:ced_db1_prod:
/data_ced1/.*.*:ced_db1_prod:

/dev/vgcedsv1/lvol1.*.*:ced_sv1_prod:
/data_ced0.*.*:ced_sv1_prod:

/dev/vgdmsdb1/.*.*:dms_db1_prod:
/data_dms1/.*.*:dms_db1_prod:

/dev/vgdwhdb1/.*.*:dwh_db1_prod:
/data_dwh1/.*.*:dwh_db1_prod:

/dev/vgdwhdb2/.*.*:dwh_db2_prod:
/data_dwh2/.*.*:dwh_db2_prod:

/dev/vgpsoftdb1/.*.*:ps_db1_prod:
/data_ps1.*.*:ps_db1_prod:

/dev/vgpsoftdb2/.*.*:ps_db2_prod:
/data_ps2.*.*:ps_db2_prod:

/dev/vgpsoftsv1/.*.*:ps_sv1_prod:
/data_ps0.*.*:ps_sv1_prod:

/dev/vgpsoftsv4.*.*:ps_sv4_prod:
/data_ps4.*.*:ps_sv4_prod:

```

```

/dev/vgppvdb1/.*.*:pv_db1_prod:
/data_pv1/.*.*:pv_db1_prod:

/dev/vgsicsdb1/.*.*:sics_db1_prod:
/data_sics1/.*.*:sics_db1_prod:

/dev/vgsicsdb2/.*.*:sics_db2_prod:
/data_sics2/.*.*:sics_db2_prod:

/dev/vgcodb1/.*.*:co_db1_prod:
/data_col/.*.*:co_db1_prod:

/dev/vgdmdb1/.*.*:dm_db1_prod:
/data_dm1/.*.*:dm_db1_prod:

```

SEE ALSO

edrcintro(1), **pkg_hostname(3)**, **getfilesystem(3)**

NOTES

The **pkg_hostname(3)** command and this configuration file was originated on HP-UX MC/ServiceGuard where the need came up to always show the user if he is in a directory related to a cluster package by changing the shell prompt.

However the **pkg_hostname(3)** is not dependent on HP-US or the MC/ServiceGuard cluster.

It can be used on any system where it is desired to show the user to which host (or alias) a directory on a certain file system is related by changing the prompt (better permanently show the user the information then relating to the idea that the user will read the documentation and knows the relationship of filesystems to applications/clusters/aliases).

BUGS

-

AUTHOR

pkg_hostname.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

zip, *zipcloak*, *zipnote*, *zipsplit* – package and compress (archive) files

SYNOPSIS

zip [-AcdDeEffghjklLmoqrSTuvVwXyz@\$] [-b *path*] [-n *suffixes*] [-t *mmddyy*] [*zipfile* [*file1* *file2* ...]] [-xi *list*]

zipcloak [-dhL] [-b *path*] *zipfile*

zipnote [-hwL] [-b *path*] *zipfile*

zipsplit [-hiLpst] [-n *size*] [-b *path*] *zipfile*

DESCRIPTION

zip is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows NT, Minix, Atari and Macintosh. It is analogous to a combination of the UNIX commands *tar*(1) and *compress*(1) and is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems).

A companion program (*unzip*(1L)), unpacks *zip* archives. The *zip* and *unzip*(1L) programs can work with archives produced by PKZIP, and PKZIP and PKUNZIP can work with archives produced by *zip*. *zip* version 2.1 is compatible with PKZIP 2.04. Note that PKUNZIP 1.10 cannot extract files produced by PKZIP 2.04 or *zip* 2.1. You must use PKUNZIP 2.04g or *unzip* 5.0p1 (or later versions) to extract them.

For a brief help on *zip* and *unzip*, run each without specifying any parameters on the command line.

The program is useful for packaging a set of files for distribution; for archiving files; and for saving disk space by temporarily compressing unused files or directories.

The *zip* program puts one or more compressed files into a single *zip* archive, along with information about the files (name, path, date, time of last modification, protection, and check information to verify file integrity). An entire directory structure can be packed into a *zip* archive with a single command. Compression ratios of 2:1 to 3:1 are common for text files. *zip* has one compression method (deflation) and can also store files without compression. *zip* automatically chooses the better of the two for each file to be compressed.

When given the name of an existing *zip* archive, *zip* will replace identically named entries in the *zip* archive or add entries for new names. For example, if *foo.zip* exists and contains *foo/file1* and *foo/file2*, and the directory *foo* contains the files *foo/file1* and *foo/file3*, then:

```
zip -r foo foo
```

will replace *foo/file1* in *foo.zip* and add *foo/file3* to *foo.zip*. After this, *foo.zip* contains *foo/file1*, *foo/file2*, and *foo/file3*, with *foo/file2* unchanged from before.

If the file list is specified as *-@*, *zip* takes the list of input files from standard input. Under UNIX, this option can be used to powerful effect in conjunction with the *find*(1) command. For example, to archive all the C source files in the current directory and its subdirectories:

```
find . -name "*.c" -print | zip source -@
```

(note that the pattern must be quoted to keep the shell from expanding it). *zip* will also accept a single dash ("-") as the *zip* file name, in which case it will write the *zip* file to standard output, allowing the output to be piped to another program. For example:

```
zip -r - . | dd of=/dev/nrst0 obs=16k
```

would write the *zip* output directly to a tape with the specified block size for the purpose of backing up the current directory.

zip also accepts a single dash ("-") as the name of a file to be compressed, in which case it will read the file from standard input, allowing *zip* to take input from another program. For example:

```
tar cf - . | zip backup -
```

would compress the output of the *tar* command for the purpose of backing up the current directory. This generally produces better compression than the previous example using the *-r* option, because *zip* can take

advantage of redundancy between files. The backup can be restored using the command

```
unzip -p backup | tar xf -
```

When no zip file name is given and stdout is not a terminal, *zip* acts as a filter, compressing standard input to standard output. For example,

```
tar cf - . | zip | dd of=/dev/nrst0 obs=16k
```

is equivalent to

```
tar cf - . | zip - - | dd of=/dev/nrst0 obs=16k
```

zip archives created in this manner can be extracted with the program *funzip* which is provided in the *unzip* package, or by *gunzip* which is provided in the *gzip* package. For example:

```
dd if=/dev/nrst0 ibs=16k | funzip | tar xvf -
```

When changing an existing *zip* archive, *zip* will write a temporary file with the new contents, and only replace the old one when the process of creating the new version has been completed without error.

If the name of the *zip* archive does not contain an extension, the extension *.zip* is added. If the name already contains an extension other than *.zip* the existing extension is kept unchanged.

OPTIONS

-A Adjust self-extracting executable archive. A self-extracting executable archive is created by prepending the SFX stub to an existing archive. The **-A** option tells *zip* to adjust the entry offsets stored in the archive to take into account this "preamble" data.

-b path

Use the specified *path* for the temporary *zip* archive. For example:

```
zip -b /tmp stuff *
```

will put the temporary *zip* archive in the directory */tmp*, copying over *stuff.zip* to the current directory when done. This option is only useful when updating an existing archive, and the file system containing this old archive does not have enough space to hold both old and new archive at the same time.

-c Add one-line comments for each file. File operations (adding, updating) are done first, and the user is then prompted for a one-line comment for each file. Enter the comment followed by return, or just return for no comment.

-d Remove (delete) entries from a *zip* archive. For example:

```
zip -d foo foo/tom/junk foo/harry/* *.o
```

will remove the entry *foo/tom/junk*, all of the files that start with *foo/harry/*, and all of the files that end with *.o* (in any path). Note that shell pathname expansion has been inhibited with backslashes, so that *zip* can see the asterisks, enabling *zip* to match on the contents of the *zip* archive instead of the contents of the current directory.

Under MSDOS, **-d** is case sensitive when it matches names in the *zip* archive. This requires that file names be entered in upper case if they were zipped by PKZIP on an MSDOS system.

-D Do not create entries in the *zip* archive for directories. Directory entries are created by default so that their attributes can be saved in the *zip* archive. The environment variable ZIPOPT can be used to change the default options. For example under Unix with sh:

```
ZIPOPT="-D"; export ZIPOPT
```

(The variable ZIPOPT can be used for any option except **-i** and **-x** and can include several options.) The option **-D** is a shorthand for **-x ""** but the latter cannot be set as default in the ZIPOPT environment variable.

-e Encrypt the contents of the *zip* archive using a password which is entered on the terminal in response to a prompt (this will not be echoed; if standard error is not a tty, *zip* will exit with an error). The password prompt is repeated to save the user from typing errors.

- f** Replace (freshen) an existing entry in the *zip* archive only if it has been modified more recently than the version already in the *zip* archive; unlike the update option (**-u**) this will not add files that are not already in the *zip* archive. For example:

```
zip -f foo
```

This command should be run from the same directory from which the original *zip* command was run, since paths stored in *zip* archives are always relative.

Note that the timezone environment variable TZ should be set according to the local timezone in order for the **-f**, **-u** and **-o** options to work correctly.

The reasons behind this are somewhat subtle but have to do with the differences between the Unix-format file times (always in GMT) and most of the other operating systems (always local time) and the necessity to compare the two. A typical TZ value is "MET-1METDST" (Middle European time with automatic adjustment for "summertime" or Daylight Savings Time).
- F** Fix the *zip* archive. This option can be used if some portions of the archive are missing. It is not guaranteed to work, so you **MUST** make a backup of the original archive first.

When doubled as in **-FF** the compressed sizes given inside the damaged archive are not trusted and *zip* scans for special signatures to identify the limits between the archive members. The single **-F** is more reliable if the archive is not too much damaged, for example if it has only been truncated, so try this option first.

Neither option will recover archives that have been incorrectly transferred in ascii mode instead of binary. After the repair, the **-t** option of *unzip* may show that some files have a bad CRC. Such files cannot be recovered; you can remove them from the archive using the **-d** option of *zip*.
- g** Grow (append to) the specified *zip* archive, instead of creating a new one. If this operation fails, *zip* attempts to restore the archive to its original state. If the restoration fails, the archive might become corrupted.
- h** Display the *zip* help information (this also appears if *zip* is run with no arguments).
- i files** Include only the specified files, as in:

```
zip -r foo . -i \*.c
```

which will include only the files that end in *.c* in the current directory and its subdirectories. (Note for PKZIP users: the equivalent command is

```
pkzip -rP foo *.c
```

PKZIP does not allow recursion in directories other than the current one.) The backslash avoids the shell filename substitution, so that the name matching is performed by *zip* at all directory levels.
- j** Store just the name of a saved file (junk the path), and do not store directory names. By default, *zip* will store the full path (relative to the current path).
- J** Strip any prepended data (e.g. a SFX stub) from the archive.
- k** Attempt to convert the names and paths to conform to MSDOS, store only the MSDOS attribute (just the user write attribute from UNIX), and mark the entry as made under MSDOS (even though it was not); for compatibility with PKUNZIP under MSDOS which cannot handle certain names such as those with two dots.
- l** Translate the Unix end-of-line character LF into the MSDOS convention CR LF. This option should not be used on binary files. This option can be used on Unix if the *zip* file is intended for PKUNZIP under MSDOS. If the input files already contain CR LF, this option adds an extra CR. This ensure that *unzip -a* on Unix will get back an exact copy of the original file, to undo the effect of *zip -l*.
- ll** Translate the MSDOS end-of-line CR LF into Unix LF. This option should not be used on binary files. This option can be used on MSDOS if the *zip* file is intended for *unzip* under Unix.

- L** Display the *zip* license.
- m** Move the specified files into the *zip* archive; actually, this deletes the target directories/files after making the specified *zip* archive. If a directory becomes empty after removal of the files, the directory is also removed. No deletions are done until *zip* has created the archive without error. This is useful for conserving disk space, but is potentially dangerous so it is recommended to use it in combination with **-T** to test the archive before removing all input files.
- n suffixes**
Do not attempt to compress files named with the given *suffixes*. Such files are simply stored (0% compression) in the output *zip* file, so that *zip* doesn't waste its time trying to compress them. The suffixes are separated by either colons or semicolons. For example:

```
zip -rn .Z:.zip:.tiff:.gif:.snd foo foo
```


will copy everything from *foo* into *foo.zip*, but will store any files that end in *.Z*, *.zip*, *.tiff*, *.gif*, or *.snd* without trying to compress them (image and sound files often have their own specialized compression methods). By default, *zip* does not compress files with extensions in the list *.Z:.zip:.zoo:.arc:.lzh:.arj*. Such files are stored directly in the output archive. The environment variable *ZIPOPT* can be used to change the default options. For example under Unix with *csh*:

```
setenv ZIPOPT "-n .gif:.zip"
```


To attempt compression on all files, use:

```
zip -n : foo
```


The maximum compression option **-9** also attempts compression on all files regardless of extension.
- o** Set the "last modified" time of the *zip* archive to the latest (oldest) "last modified" time found among the entries in the *zip* archive. This can be used without any other operations, if desired. For example:

```
zip -o foo
```


will change the last modified time of *foo.zip* to the latest time of the entries in *foo.zip*.
- q** Quiet mode; eliminate informational messages and comment prompts. (Useful, for example, in shell scripts and background tasks).
- r** Travel the directory structure recursively; for example:

```
zip -r foo foo
```


In this case, all the files and directories in *foo* are saved in a *zip* archive named *foo.zip*, including files with names starting with ".", since the recursion does not use the shell's file-name substitution mechanism. If you wish to include only a specific subset of the files in directory *foo* and its subdirectories, use the **-i** option to specify the pattern of files to be included. You should not use **-r** with the name ".*", since that matches "." which will attempt to zip up the parent directory (probably not what was intended).
- S** Include system and hidden files. This option is effective on some systems only; it is ignored on Unix.
- t mmddyy**
Do not operate on files modified prior to the specified date, where *mm* is the month (0-12), *dd* is the day of the month (1-31), and *yy* are the last two digits of the year. For example:

```
zip -rt 120791 infamy foo
```


will add all the files in *foo* and its subdirectories that were last modified on or after 7 December 1991, to the *zip* archive *infamy.zip*.
- T** Test the integrity of the new *zip* file. If the check fails, the old *zip* file is unchanged and (with the **-m** option) not input files are removed.

- u** Replace (update) an existing entry in the *zip* archive only if it has been modified more recently than the version already in the *zip* archive. For example:

```
zip -u stuff *
```

will add any new files in the current directory, and update any files which have been modified since the *zip* archive *stuff.zip* was last created/modified (note that *zip* will not try to pack *stuff.zip* into itself when you do this).

Note that the **-u** option with no arguments acts like the **-f** (freshen) option.
- v** Verbose mode or print diagnostic version info.

Normally, when applied to real operations, this option enables the display of a progress indicator during compression and requests verbose diagnostic info about zipfile structure oddities.

When **-v** is the only command line argument, and stdout is not redirected to a file, a diagnostic screen is printed. In addition to the help screen header with program name, version, and release date, some pointers to the Info-ZIP home and distribution sites are given. Then, it shows information about the target environment (compiler type and version, OS version, compilation date and the enabled optional features used to create the *zip* executable).
- V** Save VMS file attributes. This option is available on VMS only; *zip* archives created with this option will generally not be usable on other systems.
- w** Append the version number of the files to the name, including multiple versions of files. (VMS only; default: use only the most recent version of a specified file).
- x files** Explicitly exclude the specified files, as in:

```
zip -r foo foo -x \*.o
```

which will include the contents of *foo* in *foo.zip* while excluding all the files that end in *.o*. The backslash avoids the shell filename substitution, so that the name matching is performed by *zip* at all directory levels.
- X** Do not save extra file attributes (Extended Attributes on OS/2, uid/gid and file times on Unix).
- y** Store symbolic links as such in the *zip* archive, instead of compressing and storing the file referred to by the link (UNIX only).
- z** Prompt for a multi-line comment for the entire *zip* archive. The comment is ended by a line containing just a period, or an end of file condition (^D on UNIX, ^Z on MSDOS, OS/2, and VAX/VMS). The comment can be taken from a file:

```
zip -z foo < foowhat
```
- #** Regulate the speed of compression using the specified digit #, where **-0** indicates no compression (store all files), **-1** indicates the fastest compression method (less compression) and **-9** indicates the slowest compression method (optimal compression, ignores the suffix list). The default compression level is **-6**.
- @** Take the list of input files from standard input. File names containing spaces must be quoted using single quotes, as in 'file name'.
- \$** Include the volume label for the the drive holding the first file to be compressed. If you want to include only the volume label or to force a specific drive, use the drive name as first file name, as in:

```
zip -$ foo a: c:bar
```

This option is effective on some systems only (MSDOS and OS/2); it is ignored on Unix.

EXAMPLES

The simplest example:

```
zip stuff *
```

creates the archive *stuff.zip* (assuming it does not exist) and puts all the files in the current directory in it, in compressed form (the *.zip* suffix is added automatically, unless that archive name given contains a dot already; this allows the explicit specification of other suffixes).

Because of the way the shell does filename substitution, files starting with "." are not included; to include these as well:

```
zip stuff .* *
```

Even this will not include any subdirectories from the current directory.

To zip up an entire directory, the command:

```
zip -r foo foo
```

creates the archive *foo.zip*, containing all the files and directories in the directory *foo* that is contained within the current directory.

You may want to make a *zip* archive that contains the files in *foo*, without recording the directory name, *foo*. You can use the *-j* option to leave off the paths, as in:

```
zip -j foo foo/*
```

If you are short on disk space, you might not have enough room to hold both the original directory and the corresponding compressed *zip* archive. In this case, you can create the archive in steps using the *-m* option. If *foo* contains the subdirectories *tom*, *dick*, and *harry*, you can:

```
zip -rm foo foo/tom
zip -rm foo foo/dick
zip -rm foo foo/harry
```

where the first command creates *foo.zip*, and the next two add to it. At the completion of each *zip* command, the last created archive is deleted, making room for the next *zip* command to function.

PATTERN MATCHING

This section applies only to UNIX. Watch this space for details on MSDOS and VMS operation.

The UNIX shells (*sh*(1) and *csh*(1)) do filename substitution on command arguments. The special characters are:

- ?** match any single character
- *** match any number of characters (including none)
- []** match any character in the range indicated within the brackets (example: [a-f], [0-9]).

When these characters are encountered (without being escaped with a backslash or quotes), the shell will look for files relative to the current path that match the pattern, and replace the argument with a list of the names that matched.

The *zip* program can do the same matching on names that are in the *zip* archive being modified or, in the case of the *-x* (exclude) or *-i* (include) options, on the list of files to be operated on, by using backslashes or quotes to tell the shell not to do the name expansion. In general, when *zip* encounters a name in the list of files to do, it first looks for the name in the file system. If it finds it, it then adds it to the list of files to do. If it does not find it, it looks for the name in the *zip* archive being modified (if it exists), using the pattern matching characters described above, if present. For each match, it will add that name to the list of files to be processed, unless this name matches one given with the *-x* option, or does not match any name given with the *-i* option.

The pattern matching includes the path, and so patterns like **.o* match names that end in ".o", no matter what the path prefix is. Note that the backslash must precede every special character (i.e. *?*[]*), or the entire argument must be enclosed in double quotes ("").

In general, use backslash to make *zip* do the pattern matching with the *-f* (freshen) and *-d* (delete) options, and sometimes after the *-x* (exclude) option when used with an appropriate operation (add, *-u*, *-f*, or *-d*).

SEE ALSO

compress(1), shar(1L), tar(1), unzip(1L), gzip(1L)

BUGS

zip 2.1 is not compatible with PKUNZIP 1.10. Use *zip* 1.1 to produce *zip* files which can be extracted by PKUNZIP 1.10.

zip files produced by *zip* 2.1 must not be *updated* by *zip* 1.1 or PKZIP 1.10, if they contain encrypted members or if they have been produced in a pipe or on a non-seekable device. The old versions of *zip* or PKZIP would create an archive with an incorrect format. The old versions can list the contents of the *zip* file but cannot extract it anyway (because of the new compression algorithm). If you do not use encryption and use regular disk files, you do not have to care about this problem.

Under VMS, not all of the odd file formats are treated properly. Only stream-LF format *zip* files are expected to work with *zip*. Others can be converted using Rahul Dhesi's BILF program. This version of *zip* handles some of the conversion internally. When using Kermit to transfer *zip* files from Vax to MSDOS, type "set file type block" on the Vax. When transferring from MSDOS to Vax, type "set file type fixed" on the Vax. In both cases, type "set file type binary" on MSDOS.

Under VMS, *zip* hangs for file specification that uses DECnet syntax *foo::*.**.

On OS/2, *zip* cannot match some names, such as those including an exclamation mark or a hash sign. This is a bug in OS/2 itself: the 32-bit DosFindFirst/Next don't find such names. Other programs such as GNU tar are also affected by this bug.

Under OS/2, the amount of Extended Attributes displayed by DIR is (for compatibility) the amount returned by the 16-bit version of DosQueryPathInfo(). Otherwise OS/2 1.3 and 2.0 would report different EA sizes when DIRing a file. However, the structure layout returned by the 32-bit DosQueryPathInfo() is a bit different, it uses extra padding bytes and link pointers (it's a linked list) to have all fields on 4-byte boundaries for portability to future RISC OS/2 versions. Therefore the value reported by *zip* (which uses this 32-bit-mode size) differs from that reported by DIR. *zip* stores the 32-bit format for portability, even the 16-bit MS-C-compiled version running on OS/2 1.3, so even this one shows the 32-bit-mode size.

On the Amiga, the -A option currently does not work.

AUTHORS

Copyright (C) 1990-1996 Mark Adler, Richard B. Wales, Jean-loup Gailly, Onno van der Linden, Kai Uwe Rommel, Igor Mandrichenko, John Bush and Paul Kienitz. Permission is granted to any individual or institution to use, copy, or redistribute this software so long as all of the original files are included, that it is not sold for profit, and that this copyright notice is retained.

LIKE ANYTHING ELSE THAT'S FREE, ZIP AND ITS ASSOCIATED UTILITIES ARE PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

Please send bug reports and comments by email to: *zip-bugs@wkuvx1.wku.edu*. For bug reports, please include the version of *zip* (see *zip-h*), the make options used to compile it (see *zip-v*), the machine and operating system in use, and as much additional information as possible.

ACKNOWLEDGEMENTS

Thanks to R. P. Byrne for his *Shrink.Pas* program, which inspired this project, and from which the shrink algorithm was stolen; to Phil Katz for placing in the public domain the *zip* file format, compression format, and .ZIP filename extension, and for accepting minor changes to the file format; to Steve Burg for clarifications on the deflate format; to Haruhiko Okumura and Leonid Broukhis for providing some useful ideas for the compression algorithm; to Keith Petersen, Rich Wales, Hunter Goatley and Mark Adler for providing a mailing list and *ftp* site for the INFO-ZIP group to use; and most importantly, to the INFO-ZIP group itself (listed in the file *infozip.who*) without whose tireless testing and bug-fixing efforts a portable *zip* would not have been possible. Finally we should thank (blame) the first INFO-ZIP moderator, David Kirschbaum, for getting us into this mess in the first place. The manual page was rewritten for UNIX by R. P. C. Rodgers.

NAME

pid – Invoke ps With The "Right" Switches and grep For a Pattern

SYNOPSIS

edrc/bin/pid [*process-name*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

invokes a **ps** with the "right" switches and makes a **grep** for the given pattern. The whole line of **ps** is printed to standard output.

OPTIONS

process-name

Process name which has to be checked.

EXIT STATUS

0 always

EXAMPLES

-

SEE ALSO

edrcintro(1), **ps(1)**, **pslist(3)**, **grep(1)**, **xpid(1)**

FILES

-

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

BUGS

-

AUTHOR

pid was developed by Peter Stevens, SFI and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Name

pl

Synopsis

```
pl -prefab prefabname parm=value .. [-options]
.. OR ..
pl scriptfile [-options]
```

Description

pl is the primary component of the 'ploticus' data display package

pl is a program that produces plots and charts from data, and produces results that can be viewed on web pages, paper, slides, or interactively on the screen. Standard types of plots may be done using *prefab* plot templates, or a user-developed *script* file may be supplied for greater flexibility and customization. **pl** may be executed from the command line or as a CGI program.

For complete online docs and downloads see <http://ploticus.sourceforge.net>

Where to find examples

See the various *prefab* examples. A large number of *script* examples are also available. Some usage examples are also shown below.

Command line arguments

Command line arguments may generally be given in any order. If there are arguments that you want to always have in effect, you can invoke them from a *config* file. Many settings can also be made dynamically from scripts via *proc settings* or *proc page*. Processing occurs in this order: first the *config* file is read; then command line args are processed (left to right); then *proc page* and/or *proc settings*. Later settings override earlier ones.

Basic command line options

-prefab *prefabname*

Produce a plot using a *prefab* plot template. *prefabname* identifies the template, eg. *cron* or *vbars*. Necessary parameters are supplied on the command line using the form *parm=value*.

scriptfile

-f *scriptfile*

names a *script* file that will be interpreted to produce results. Alternatively, **-stdin** may be used to indicate that script will be available on standard input.

variable=value

Declares the named *variable* and sets it to the given *value*. This is a convenient way to pass information to *prefabs* and *scripts*. Variable names are case-sensitive.

Example: CUTDATE=10-31-98

sets the variable CUTDATE to 10-31-98.

-o *outfile* | *stdout*

Specify a filename where the result will be written. No processing is applied to this name.. so the ending should be appropriate for the selected output format, eg. use `.png` for PNG files. If `-o stdout` is used, result will be sent to standard output. If **-o** is not specified, a default output filename will be used.

Example: `-o fp001.png`

-dir *dirname*

Set ploticus' working directory to *dirname*. If used, this argument should be specified leftmost on the command line, since it affects evaluation of other args.

Result format options

(Availability depends on your ploticus configuration/build)

-png PNG image

-gif pseudo-GIF image

-jpeg JPEG image

-svg or **-svgz** SVG graphic. See also SVG / XML options below.

-swf SWF (flash) result.

-wbmp WBMP image

-eps EPS (encapsulated PostScript)

-ps paginated PostScript to stdout

-x11 display on X11 screen

-drawdump *filename* produce no visible graphic; save a generic representation of the graphic result to a file. By using `-drawdump` and `-drawdumpa` you can easily overlay or combine results from separate pl runs. The drawdump file can be rendered later in any desired format, using this command: `pl -pre-fab draw dumpfile=filename` or by using `proc draw` commands. Drawdump capability is available in all builds. (2.30+)

-drawdumpa *filename* same as `-drawdump` but result is **appended** to file.

Clickable image maps and mouseover options

-csmmap

produce a client-side clickable imagemap to accompany a png, gif, or jpeg. These can be used for hyperlinks, and also for providing pop-up text labels that appear when the mouse passes over a region. By default, client-side map content is written to stdout.

-csmmapdemo

Same as **-csmmap** but all mapped regions are shown outlined in green, and a complete HTML chunk is produced which involves the output image name.

-mapfile *filename* | stdout | stderr

explicitly name the output file containing the map info. The name may also be set in `proc` page. If a name is not specified, client-side image map info will be written to stdout; For SVG this parameter is not needed, since image map info is embedded in the SVG file.

-map

produce a server-side clickable imagemap file to accompany a png, gif, jpeg, or SVG. **Note: server-side maps are deprecated.**

Result sizing options

-scale *sx[,sy]*

Scale the final result. If one value is given, the result is scaled by this amount in both x and y. If two values are given, scaling in x and scaling in y may be done independently. A scale value of less than 1.0 reduces the size; an scale value of greater than 1.0 enlarges. Scaling is done relative to the origin (0,0) which is at the lower left.

Example: `-scale 0.7`

-pagesize *width,height*

Sets the pre-crop size of the result image for GIF/PNG/JPEG, or sets the display window size when drawing to X11. On other output devices this option does nothing. *width* and *height* are in absolute units. 0,0 is the lower left corner. If `-pagesize` is not specified, the default size will be 8" x 8". Size is set before any drawing takes place and is unaffected by the `-scale` option.

When rendering PNG/GIF/JPEG images, this option determines amount of internal memory allocation for accommodating the image. The result can never be bigger than this size, and any drawing outside the bounds will not be visible. **To create PNG/GIF/JPEG images larger than 8" x 8", this option MUST be specified to set a bigger size.** Cropping options (below) can be used along with `-pagesize` as long as they result in a smaller rectangle than the `pagesize`; they take effect after all drawing has been completed.

-pagesize has no effect with EPS or paginated PostScript results (the PostScript BoundingBox will be determined by the extent of the graphic).

Example: `-pagesize 7,3`

-tightcrop

For image or EPS output, crop the result tightly to the extent of the design. Normally a small margin is allowed on all four sides. This option sometimes crops a bit too tight; if so try **-croprel**.

-crop *x1,y1,x2,y2*

Crop image or EPS result to the box specified by *x1,y1* and *x2,y2*, in absolute units.

Note that there may be no spaces in the coordinates specification. Cropping takes place after design is rendered and does not affect coordinate locations.

Example: `-crop 1.2,0.8,4.4,5.2`

-croprel *left,bottom,right,top*

Crop image or EPS result tightly to the extent of the design (like **-tightcrop**), but then adjust the cropping outward or inward on one or more sides. *left* is the amount to adjust the left side, in absolute units. Similarly for *bottom*, *right*, and *top*. Positive values always adjust outward from center; negative values adjust inward (tighter). There may be no spaces in the *left,bottom,right,top* specification. Cropping takes place after design is rendered and does not affect coordinate locations.

Example: `-croprel 0,-0.1,0,0.1`

-pixsize *width,height*

If specified, result PNG/GIF/JPG image will be created at exactly this width and height in pixels. Does not interact with scaling or cropping... user is responsible for ensuring that content fits appropriately into the specified size. User is also responsible for setting `-pagesize` appropriately for larger images. New in 2.40

Graphics environment options

-font *font*

sets the overall font to *font*. See `fonts` for more info.

-textsize *pointsize*

sets the overall default textsize to *pointsize*. All embedded size specifications will be rendered relative to this.

-linewidth *w*

sets the overall default linewidth to *w*. All embedded line width specifications will be rendered relative to this. See `linedetails(pli)` for more on line width.

-color *color*

sets the overall default text and line drawing color to *color*.

-backcolor *color*

sets the background color to *color*.

-cm

Use centimeters as your absolute units, instead of inches. On the command line this must appear to the left of any arguments dealing with absolute unit values, such as `-pagesize`. Centimeter absolute units can also be set via `proc` settings. If `cm` will always be the desired absolute units, the preferred way to achieve this is by using `units: cm` in a `ploticus` config file.

-inches

Use inches as your absolute units. This is the default.

-outlabel *label*

Set the label or title for the output. For X11 this sets the window title; for PostScript and SVG it sets the `%%Title` attribute.

Capacity setting options

These options (new with version 2.10) allow capacities to be raised for accomodation of very large data sets, or lowered to minimize memory usage. The defaults in this section are defined in `pl.h`.

-maxrows *nrows*

Set the capacity for data rows to *nrows*. Default *nrows* is 10,000. Ploticus will allocate one pointer for each row.

-maxfields *nfields*

Set the capacity for data fields to *nfields*. Default *nfields* is 200,000. Ploticus will allocate one pointer for each field.

-maxproclines *nlines*

Set the capacity for script lines for active procs to *nlines*. Default *nlines* is 5000. Active procs are the current proc, all `#saved` procs, and all `proc getdata` procs that contain embedded data. Ploticus will allocate one pointer for each line in each active proc.

-maxvector *ncells*

Set the capacity for the data plotting vector to *ncells*. Default *ncells* is 100,000. The data plotting vector is an array which holds plottable values for situations where the values must be sorted or pre-screened for bad values. Ploticus will allocate one `double` value for each cell.

-maxdrawpoints *n*

Use this if you need to render a polygon having more than 500 points in PNG/GIF/JPEG, X11, or SWF, or any continuous line having more than 500 points in SWF.

Note: raising the maximum number of categories may be done using `proc categories` from within the script.

-cpulimit #Include nbsp2 *s*

Set unix resource limit on cpu time to *s* seconds. Default is 30 seconds. New in 2.40

SVG / XML options:

-svg_tagparms *string*

This allows arbitrary `text` to be inserted into the opening `<svg>` tag.

Example: `-svg_tagparms 'height="10cm" width="15cm"'`

-omit_xml_declaration

By default the first line of the SVG result will be the XML declaration `<?xml . . >`. Use this option to suppress the XML declaration line if the SVG result is to be embedded into a larger XML document.

-xml_encoding *method*

Set the XML character encoding method. This encoding will be indicated in the XML declaration line. The default is `iso-8859-1` which provides Latin and Western European character sets. For Unicode fonts this should be set to `utf-8` (for more discussion see the Unicode section in fonts).

-tag

Causes a suitable HTML `<EMBED>` tag to be written to standard output.

-zlevel *n*

This may be used to set the compression level to *n* for SVGZ output (0 - 9 where 9 is highest level of compression and the default).

Interactive (workstation) use options

-winloc *x,y*

Control where on the screen the upper-left corner of the X11 display window will be placed. *x* and *y* are in pixels. Example: `-winloc 200 0`

-v *command*

-viewer *command*

After generating results in the specified format, execute *command* in order to view the results on your screen. The output file will automatically be included in the *command*. For example, if a GIF file is being generated you might use this to invoke the `xv` utility: `-viewer xv`. If

PostScript is being generated you could use something like this to invoke the ghostview utility: `-viewer "gv -magstep -1"`. The given command must be available on your system and locatable in your command search path. This option may not be used with `-o stdout`.

-noshell

If specified, ploticus is prohibited from issuing any shell commands. This is a security feature useful for example when running a script that was sent to you by an unknown party. New in 2.31

Paper orientation options

-landscape

For paginated postscript, set paper orientation to landscape (oblong).

-portrait

For paginated postscript, set paper orientation to portrait.

-posteroffset *x,y*

Allows production of large-size posters made up of multiple standard sheets of paper butted together. May be used only with paginated PostScript, and should be used in combination with the `-scale` and `-textsize` options. *x,y* is the point within your result (in absolute units) that is to be placed at the lower left corner of the page. For further discussion of this, see `posters`.

Development and debugging options

-debug

Debug mode. Causes diagnostic information to be written to the diagnostic stream (stderr by default, see `-diagfile` below). Highly recommended if you are experiencing difficulty. Best to use `-debug` as the first (leftmost) argument so that it can report on all arguments gotten. Another effect of debug mode is that any temporary files are not removed upon termination.

-ping

Write the ploticus name and version number to standard output and exit. versions 2.33+

-echo [*diag* | *stdout*]

Write ploticus script lines as they are executed. Lines are written to the diagnostic stream (standard error by default) or standard output. Lines are written after variables and most script directives, including flow-of-control directives, are evaluated.

-showbad

Identify unplottable data, showing the value, and its row and field.

-diagfile *filename* | *stderr* | *stdout*

All non-error messages and output will be written to this file (default is stderr).

-errmsgpre *tag*

Allows developer to set the first portion of all ploticus error messages to *tag* for purposes of presentation or identification.

-errfile *filename* | *stderr* | *stdout*

All error messages will be written to this file (default is stderr).

-help or **-?** or **-version**

Print version number, copyright info, web site address, etc.

Output file names

The output file may be specified on the command line using the **-o** option, or via Proc Page's `outfilename` attribute. If so, the result is written to a file of that name. `-o stdout` may also be used to send result to standard output.

Otherwise, if your script filename has a "recognized extension" (**.p**, **.pl**, **.plo**, **.pls**, **.htm** or **.html**), the base part of the script file name is used and **.png**, **.gif**, etc. is appended. If your script filename doesn't have a recognized extension, the generic name `out.*` will be used.

X11 output is always displayed on the screen, and paginated PostScript is written to standard output unless `-o` is used.

If page breaks (Proc Page) are encountered when rendering in any format other than paginated PostScript, special action is necessary since each page must go into a separate file. A Proc Page `outfilename` may be specified for each page; otherwise a `pn` prefix will be attached to the beginning of each page's output file name to indicate page *n*.

If a `clickmap` is being generated, the result file is named similarly to the above.

Usage examples

The following example uses the `scat` prefab:

```
pl -prefab scat -png datafile=results.dat x=2 y=3
```

The following examples assume that you have a script file called `lineplot1.p`.

```
pl -x lineplot1.p = view on X11 screen
pl -png lineplot1.p = create PNG image lineplot1.png
pl -gif lineplot1.p -o stdout = create GIF image on standard output
pl -gif lineplot1.p -viewer xv = produce GIF and view using xv (assuming xv
image viewer is available on your system).
```

```

pl -eps lineplot1.p = produce EPS file lineplot1.eps
pl -eps lineplot1.p -viewer gv = produce EPS and view using gv (that's ghostview,
assuming it is available on your system).
pl -eps lineplot1.p -o lineplot.eps = produce EPS into file lineplot.eps
pl -ps lineplot1.p | lp = produce paginated postscript and send to unix lp print
spooler.
pl -ps lineplot1.p -viewer gv = produce paginated postscript and view using ghost-
view.

```

Environment

PLOTICUS_CONFIG

The name of a ploticus configuration file, for setting default date notations, number notations, measurement units, etc.

PLOTICUS_PREFABS

The path name of a directory where ploticus will look for prefab scripts. The "factory" prefabs are located in the ploticus ./prefabs subdirectory.

LC_CTYPE, LC_COLLATE, LANG

Locale support. Thanks to Oleg Bartunov oleg@sai.msu.su for contributing this. **pl** must be built with -DLOCALE for this to work.

TDH_ERRMODE

Control the disposition of error messages. Allowable values: `stderr` which is the default, and `cgi` which causes error messages to be written to stdout with html formatting.

Bugs

Ploticus has some stated limitations (mostly related to capacities that you may run into if you're dealing with large data sets). To report problems or get help see the ploticus support page.

Author, Copyright, Licensing

The primary author is Stephen C. Grubb. Ploticus covered by the General Public License (GPL)... please see the ploticus copyright page for more info.

See also

<http://ploticus.sourceforge.net>

NAME

pmdesc – List name, version, and description of all installed perl modules and PODs

SYNOPSIS

```
pmdesc [-h]

pmdesc [-s] [-t ddd] [-v dd] [--] [dir [dir [dir [...]]]]

pmdesc [-s] [-d] [--] [dir [dir [dir [...]]]]
```

DESCRIPTION

Find name, version and description of all installed Perl modules and PODs. If no directories given, searches @INC. The first column of the output (see below) can be used as module name or FAQ-name for perldoc.

Some modules are split into a pm-file and an accompanying pod-file. The version number is always taken from the pm-file.

The description found will be cut down to a length of at most 150 characters (prevents slurping in big amount of faulty docs).

Output

The output looks like this:

```
:
:
IO::Socket          (1.28)  Object interface to socket communications
IO::Socket::INET    (1.27)  Object interface for AF_INET domain sockets
IO::Socket::UNIX    (1.21)  Object interface for AF_UNIX domain sockets
IO::Stty            (n/a)   <description not available>
IO::Tty             (1.02)  Low-level allocate a pseudo-Tty, import constants.
IO::Tty::Constant   (n/a)   Terminal Constants (autogenerated)
:
:
```

The three parts module name, version and description are separated by at least one blank.

OPTIONS

-h print help message; show search path

-s sort output (not under Windows)

-d csv output

-t ddd name column has width ddd (1–3 digits); default 52

-v dd version column has width dd (1–2 digits); default 16

REQUIREMENTS

ExtUtils::MakeMaker, File::Find, Getopt::Std

BUGS AND LIMITATIONS

The command line switch `-s` (sort) is not available under non-UNIX operating systems. An additional shell sort command can be used.

There are no known bugs in this module.

Please report problems to Fritz Mehner, mehner@fh-swf.de.

AUTHORS

Tom Christiansen, tchrist@perl.com (pmdesc)

Aristotle, <http://qs321.pair.com/~monkads/> (pmdesc2)

Fritz Mehner, mehner@fh-swf.de (pmdesc3.pl)

SEE ALSO

[edrcintro\(1\)](#), [csv\(3\)](#), [lspm\(1\)](#), [perlenv\(3\)](#), [perl_modules\(3\)](#)

NOTES

pmdesc is based on pmdesc2 (Aristotle, <http://qs321.pair.com/~monkads/>). pmdesc adds extensions and bugfixes.

pmdesc2 is based on pmdesc (Perl Cookbook, 1. Ed., recipe 12.19). pmdesc2 is at least one magnitude faster than pmdesc.

VERSION

1.2.3

NAME

pod2html – convert .pod files to .html files

SYNOPSIS

```
pod2html --help --htmldir=<name> --htmlroot=<URL>
        --infile=<name> --outfile=<name>
        --podpath=<name>:...:<name> --podroot=<name>
        --cachedir=<name> --flush --recurse --norecurse
        --quiet --noquiet --verbose --noverbose
        --index --noindex --backlink --nobacklink
        --header --noheader --poderrors --nopoderrors
        --css=<URL> --title=<name>
```

DESCRIPTION

Converts files from pod format (see `perlpod`) to HTML format.

ARGUMENTS

pod2html takes the following arguments:

help

 --help

Displays the usage message.

htmldir

 --htmldir=name

Sets the directory to which all cross references in the resulting HTML file will be relative. Not passing this causes all links to be absolute since this is the value that tells Pod::Html the root of the documentation tree.

Do not use this and `--htmlroot` in the same call to pod2html; they are mutually exclusive.

htmlroot

 --htmlroot=URL

Sets the base URL for the HTML files. When cross-references are made, the HTML root is prepended to the URL.

Do not use this if relative links are desired: use `--htmldir` instead.

Do not pass both this and `--htmldir` to pod2html; they are mutually exclusive.

infile

 --infile=name

Specify the pod file to convert. Input is taken from STDIN if no infile is specified.

outfile

 --outfile=name

Specify the HTML file to create. Output goes to STDOUT if no outfile is specified.

podroot

 --podroot=name

Specify the base directory for finding library pods.

podpath

 --podpath=name:...:name

Specify which subdirectories of the podroot contain pod files whose HTML converted forms can be linked-to in cross-references.

cachedir

--cachedir=name

Specify which directory is used for storing cache. Default directory is the current working directory.

flush

--flush

Flush the cache.

backlink

--backlink

Turn =head1 directives into links pointing to the top of the HTML file.

nobacklink

--nobacklink

Do not turn =head1 directives into links pointing to the top of the HTML file (default behaviour).

header

--header

Create header and footer blocks containing the text of the “NAME” section.

noheader

--noheader

Do not create header and footer blocks containing the text of the “NAME” section (default behaviour).

poderrors

--poderrors

Include a “POD ERRORS” section in the outfile if there were any POD errors in the infile (default behaviour).

nopoderrors

--nopoderrors

Do not include a “POD ERRORS” section in the outfile if there were any POD errors in the infile.

index

--index

Generate an index at the top of the HTML file (default behaviour).

noindex

--noindex

Do not generate an index at the top of the HTML file.

recurse

--recurse

Recurse into subdirectories specified in podpath (default behaviour).

norecurse

--norecurse

Do not recurse into subdirectories specified in podpath.

css

--css=URL

Specify the URL of cascading style sheet to link from resulting HTML file. Default is none style sheet.

title

--title=title

Specify the title of the resulting HTML file.

quiet

--quiet

Don't display mostly harmless warning messages.

noquiet

--noquiet

Display mostly harmless warning messages (default behaviour). But this is not the same as "verbose" mode.

verbose

--verbose

Display progress messages.

noverbose

--noverbose

Do not display progress messages (default behaviour).

AUTHOR

Tom Christiansen, <tchrist@perl.com>.

BUGS

See Pod::Html for a list of known bugs in the translator.

SEE ALSO

perlpod, Pod::Html

COPYRIGHT

This program is distributed under the Artistic License.

NAME

portscan – scan for open ports on host from min to max

SYNOPSIS

edrc/lib/portscan [**-h**]

portscan *host* [*min* [*max*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

scan a target system for active ports.

It is possible to specify the minimum port where the scan should start and the maximum port number where the scan should stop.

Ports that are active are returned to **stdout**.

The port is scanned by connecting to the target system using the given port using a timeout of 1 second.

OPTIONS

-h print usage message.

host hostname.

min minimum port number.

If the minimum port number is not given, the default of 0 is used.

max maximum port number.

If the maximum port number is not given, the default of 65000 is used.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|---------------------------|
| 0 | no error. |
| 1 | cannot resolve host name. |
| 2 | cannot create socket. |
| 4 | usage message printed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **gethostbyname(1)**, **ping(8)**

NOTES

-

BUGS

sometimes open ports are not reported.

IPv6 is not supported yet.

AUTHOR

portscan was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

print_header – print a standard report header

SYNOPSIS

edrc/lib/print_header [**-h**]

print_header ["*report title*" [*indent* [*add_hostname*]]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print a standard report header to the terminal. The report header consists of the report title, the customer, the environment and the date of the report creation.

The customer and the environment part are resolved by **server_environment(3)** internally.

The output is limited to the current terminal width if the environment variable **\$PRINT_FIT2WIDTH** is not set to *False*.

OPTIONS

-h usage message.

"report title"
title of the report. The specified text is converted to upper case.

indent left indent of the output in characters. If not specified the output is not indented.

add_hostname
set to **True** to add a **Hostname:** ... entry after the **Environment:** ... line.

ENVIRONMENT

\$PRINT_FIT2WIDTH

If this environment variable is not set to *False*, the output is limited to the current width of the terminal window and rows extending the window width are marked with '>>'.

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

1) print a user account report to the terminal

This report is intended to be viewed on the terminal. All lines extending the terminal window width will be cut an marked with '>>' to show that there is hidden information.

```
#!/bin/sh
:
:
print_header "LOCAL USER ACCOUNTS"

{
    echo USERNAME:PASSWD:UID:GID:COMMENT:HOMEDIR:SHELL
    cat /etc/passwd
} | print_list ":"
:
:
```

The output of this example will look similar to:

```
Report:      LOCAL USER ACCOUNTS
Customer:    ACME
Environment: [PRODUCTION] ACME Corporation, Swit >>
Date:        Sun Jan 11 19:44:53 MET 2009

USERNAME     PASSWD  UID   GID   COMMENT  >>
-----
root         x       0     3     root     >>
bin          x       1     1     bin      >>
daemon      x       2     2     daemon   >>
adm          x       3     4     adm      >>
lp          x       4     7     lp        >>
sync        x       5     0     sync     >>
shutdown    x       6     0     shutdown >>
halt        x       7     0     halt     >>
(8)
```

2) create a user account report to be saved to a file

This report is intended to be saved to a file. This is why the environment variable **\$SPRINT_FIT2WIDTH** is set to *False* in this example.

```
#!/bin/sh
:
:
PRINT_FIT2WIDTH=False; export PRINT_FIT2WIDTH
print_header "LOCAL USER ACCOUNTS"

{
    echo USERNAME:PASSWD:UID:GID:COMMENT:HOMEDIR:SHELL
    cat /etc/passwd
} | print_list ":"
:
:
```

SEE ALSO

edrcintro(1), **csv(3)**, **csvcat(3)**, **print_list(3)**, **print_index(3)**, **select_columns(3)**

NOTES

The four commands **print_header(3)**, **select_columns(3)**, **print_header(3)** and **print_index(3)** provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

print_header was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

print_index – print a standard column index

SYNOPSIS

edrc/lib/print_index [-h]

print_index ["*field separator*" [*indent* [*columns* [*fixwidth* [*nospace*]]]]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print a standard report column index to the terminal.

The output is limited to the current terminal width if the environment variable **\$PRINT_FIT2WIDTH** is not set to *False*.

OPTIONS

-h usage message.

"field separator"

field separator of the input data given via **stdin**. If not specified, the field separator defaults to the semicolon (;).

indent left indent of the output in characters. If not specified the output is not indented.

columns number of columns of the column index output. If not specified a default of 2 applies.

fixwidth set to **False** for dynamic description column with, else (**True**) all columns have the identical width.

nospace set to **True** to eliminate tailing spaces at index lines, else (**False**) all lines have the identical width filled with tailing spaces.

ENVIRONMENT

\$PRINT_FIT2WIDTH

If this environment variable is not set to *False*, the output is limited to the current width of the terminal window and rows extending the window width are marked with '>>'.

EXIT STATUS

| | |
|----------|------------------|
| 0 | no error. |
| 4 | usage displayed. |

FILES

-

EXAMPLES**1) print a column index to the terminal**

This report is intended to be viewed on the terminal. All lines extending the terminal window width will be cut and marked with '>>' to show that there is hidden information.

```
#!/bin/sh
:
:
cat<<EOM | print_index
ZIP;Postal Zip Code
CITY;Name of the City
STATE;State
CNT;Country Shortcut
COUNTRY; Country Full Name
EOM
:
:
```

Result of this script when the terminal window width is smaller than the output width:

COLUMN INDEX:

```
ZIP ..... Postal Zip Code      CNT ..... >>
CITY ..... Name of the City    COUNTRY .. >>
STATE ..... State
```

2) print a column index to be saved to a file

This report is intended to be saved to a file. This is why the environment variable **\$PRINT_FIT2WIDTH** is set to *False* in this example.

```
#!/bin/sh
:
:
PRINT_FIT2WIDTH=False; export PRINT_FIT2WIDTH

cat<<EOM | print_index
ZIP;Postal Zip Code
```

```
CITY;Name of the City
STATE;State
CNT;Country Shortcut
COUNTRY; Country Full Name
EOM
:
:
```

SEE ALSO

edrcintro(1), **print_header(3)**, **print_list(3)**, **select_columns(3)**

NOTES

The four commands **print_index(3)**, **select_columns(3)**, **print_index(3)** and **print_index(3)** provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

print_index was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

print_list – format CSV data to a list with dynamic column widths

SYNOPSIS

edrc/lib/print_list [-h]

print_list ["*field_separator*" [*indent*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

the intention of this command is to print a list to a terminal in a good readable format. A CSV data provided via pipe from **stdin** is formatted and printed to **stdout**.

The first row is treated as header row. The width of all columns is dynamically adjusted to the row containing the longest column entry.

Text columns are aligned to the left, bare number columns are aligned to the right.

Furthermore the output is limited to the current terminal width if the environment variable **\$PRINT_FIT2WIDTH** is not set to *False*.

To select named columns from the CSV data prior to the formatting of the list with **print_list** use **select_columns(3)**.

OPTIONS

-h usage message.

"*field_separator*"
field separator. If not specified, the default field separator is the semicolon (;).

indent left indent of the output in characters. If not specified the output is not indented.

ENVIRONMENT

\$PRINT_FIT2WIDTH

If this environment variable is not set to *False*, the output is limited to the current width of the terminal window and rows extending the window width are marked with '>>'.

EXIT STATUS

| | |
|----------|------------------|
| 0 | no error. |
| 4 | usage displayed. |

FILES

-

EXAMPLES**1) print CSV data as formatted list**

The following CSV data received via pipe

```
cat<<EOM | print_list
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

results in the output:

| ZIP | CITY | STATE | CNT | COUNTRY |
|-------|-----------|-------|-----|--------------------------|
| ---- | ----- | ----- | --- | ----- |
| 93117 | Goleta | CA | USA | United States of America |
| 8222 | Beringen | SH | CH | Switzerland |
| 8005 | Cape Town | WC | RSA | South Africa |

(3)

2) print selected columns of CSV data as formatted list

The columns CNT, ZIP and CITY of the CSV data are selected using the **select_columns(3)** command and then formatted to a list using the **print_list** command

```
cat<<EOM | select_columns ";" "CNT;ZIP;CITY" | print_list
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

what will result in the output:

| CNT | ZIP | CITY |
|-----|-------|----------|
| --- | ----- | ----- |
| USA | 93117 | Goleta |
| CH | 8222 | Beringen |

print_list(3)

Library Commands

print_list(3)

RSA 8005 Cape Town
(3)

SEE ALSO

edrcintro(1), **csv(3)**, **csvcat(3)**, **print_header(3)**, **print_index(3)**, **select_columns(3)**

NOTES

The four commands **print_header(3)**, **select_columns(3)**, **print_list(3)** and **print_index(3)** provide the functionality to efficiently produce ASCII reports having an identical look.

BUGS

-

AUTHOR

print_list was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

version, msg – general program functions

SYNOPSIS

```
#define PROGNAME "mycommand"
#define PROGVERSION "1.0.07"
#define PROGAUTHOR "John Doe"

#include "program.h"

void version();

void msg(char *level, char *format, ...);
```

AVAILABILITY

WA2L/edrc

DESCRIPTION

program.h provides general program functions.

msg()

return a structured message to **stderr**.

Example:

```
msg("error", "cannot open file '%s' defined in line number '%d'\n", filename,
```

Output:

```
mycommand-ERROR: cannot open file 'config.cfg' defined in line number '16'
```

version()

print program version information.

The program name, version and author is printed from the **PROGVERSION**, **PROGNAME**, and **PROGAUTHOR** defines.

The output of the program name and version is printed **stdout**, the program author is printed to **stderr**.

Example:

```
version();
```

Output:

```
mycommand 1.0.07 John Doe
```

RETURN VALUE

-

ENVIRONMENT

-

FILES

`lib/$OSID/includes/program.h`

EXAMPLES

-

SEE ALSO

`edrcintro(1)`, <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>, `checkopt.h(3)`, `osid(3)`, `strings.h(3)`, `utility.h(3)`, `wa2lc(3)`

NOTES

-

BUGS

-

AUTHOR

`program.h` was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pscount – count operating systems processes

SYNOPSIS

edrc/lib/pscount [**-h**]

cat *ps_list* | **pscount**

pscount < *ps_list*

cat *ps_list* | **pscount** [**-l**] [**-d** *delimiter*] [**-i** *indent*] [**-f** *processfile*]

cat *ps_list* | **pscount** { **-s** | **-u** } [**-d** *delimiter*] [**-f** *processfile*]

pscount -p [**-d** *delimiter*] [**-f** *processfile*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

count processes and evaluate if the situation correlates to a defined situation as received thru **stdin**.

This command is intended to be used to check if the correct number of needed processes for an application are running on the system.

The format of the process count selection definition *ps_list* is:

MIN;*MAX*;*DESCRIPTION*;*USER*;*PROCESS_SELECTION*;

where

MIN Minimum number of processes that are expected (integer number). If there is no defined minimum, this field should be left empty. It is allowed to use spaces in the *MIN* field to align the numbers.

MAX Maximum number of processes that are expected (integer number). If there is no defined maximum, this field should be left empty. It is allowed to use spaces in the *MAX* field to align the numbers.

DESCRIPTION

Description of the selected processes.

USER

Username of the selected processes. This is a regular expression. If there is no defined user, this field can be left empty.

PROCESS_SELECTION

Process selection. This is a regular expression. When using the **pscount -p** command to get a first start for the *ps_list*, be aware, that there might be needed some more escaping of regular expression terms in addition to the ones given by the command output.

OPTIONS

- h** usage message.
- l** long output. Beside the **DESCRIPTION**, **MIN**, **MAX**, **CURRENT** and **STATE** columns, the **USER** and **PROCESS_SELECTION** columns are also printed.
- u** print processes not matched by any count selection. This is a verification function to help to check if all processes that relate to an application are selected in the process selection list.
- s** silent, no text output of counting result, only set the **EXIT STATUS**.
- p** print process list based on system. This output can be used to generate a process count selection definition or a *filename.kp* (known process) file.

The header line 'MIN;MAX;DESCRIPTION;USER;PROCESS_SELECTION;' can be kept in the **.kp* file, it does not influence the counting.

To print a nice formatted output of the 'known process' file, the command **pscount -p | csv list** can be used.
- d *deliminator*** record deliminator.
- i *indent*** left indent of output.
- f *processfile*** read the processes from a file instead directly from the system.

The format of the process file is the output created by the command **ps -eww -o user,command** on Linux and **ps -e -o user,command** on other operating systems.

ENVIRONMENT

-

EXIT STATUS

- 0** number of processes on the system are as defined.
- 1** number of processes on the system are different from the provided definition.

- 2 Operating system not supported. See **osid(3)** if you get this error.
- 4 usage printed.
- 5 program aborted.
- 11 a temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

edrc/var/pscount/

base directory for the 'known process' files. The ***.kp** files are read when using the **pscount -p** command.

edrc/var/pscount/apps/<Application>.kp

in the **app** directory, application related ***.kp** files are saved.

edrc/var/pscount/os/<OSID>.kp

in the **os** directory, operating system related ***.kp** files are saved. The filename must be as returned by the **osid(3)** command.

edrc/var/pscount/template.kp

template ***.kp** file to be used for application or operating system 'known process' files. To efficiently create a 'known process' file of for a running system, use the **pscount -p > filename.kp** command and edit the *filename.kp* file.

EXAMPLES

1) count processes using the 'in here' mechanism (e.g. in a script)

```
cat << EOM | pscount
1; 1;Oracle Listener ;oracle;./tnslsnr LISTENER .+;
1;10;Oracle DB Writer;oracle;ora_dbw[0-9]_ACMEDB;
1; ;Oracle Connects ;oracle;oracleACMEDB .+;
EOM
```

| DESCRIPTION | MIN | MAX | CURRENT | STATE |
|------------------|-----|-----|---------|-------|
| ----- | --- | --- | ----- | ----- |
| Oracle Listener | 1 | 1 | 1 | OK |
| Oracle DB Writer | 1 | 10 | 2 | OK |
| Oracle Connects | 1 | | 0 | FAIL |
| *TOTAL* | 3 | 11 | 3 | FAIL |

(4)

2) count processes based on an input file

Process count definition list saved in a file:

```
cat ACME_DBs.kp
```

```
#
# ACME_DBs.kp - Process List for ACME Databases
#
# [00] 20.12.2012 CWa Initial Version
#
1; 1;Oracle Listener ;oracle;./tnslsnr LISTENER .+;
1;10;Oracle DB Writer;oracle;ora_dbw[0-9]_ACMEDB;
1; ;Oracle Connects ;oracle;oracleACMEDB .+;
```

Use the process count definition list thru a pipe:

```
cat ACME_DBs.kp | pscount
```

| DESCRIPTION | MIN | MAX | CURRENT | STATE |
|------------------|-----|-----|---------|-------|
| Oracle Listener | 1 | 1 | 1 | OK |
| Oracle DB Writer | 1 | 10 | 2 | OK |
| Oracle Connects | 1 | | 5 | OK |
| *TOTAL* | 3 | 11 | 8 | OK |

or use the process count definition list thru input redirection:

```
pscount -l < ACME_DBs.kp
```

| DESCRIPTION | MIN | MAX | CURRENT | STATE | USER | PROCESS_SELECTION |
|------------------|-----|-----|---------|-------|--------|----------------------|
| Oracle Listener | 1 | 1 | 1 | OK | oracle | ./tnslsnr LISTENER . |
| Oracle DB Writer | 1 | 10 | 2 | OK | oracle | ora_dbw[0-9]_ACMEDB |
| Oracle Connects | 1 | | 5 | OK | oracle | oracleACMEDB .+ |
| *TOTAL* | 3 | 11 | 8 | OK | | |

SEE ALSO
edrcintro(1), csv(3), osid(3), pslist(3)

NOTES
The **pscount** command uses the **pslist(3)** command internally to resolve the processes.

BUGS
-

AUTHOR
pscount was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2022 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pscount.kp – file format for 'known process' files and process selection definition for pscount

SYNOPSIS

edrc/var/pscount/*/<filename>.kp

AVAILABILITY

WA2L/edrc

DESCRIPTION

See manual page **pscount(3)**.

FILEFORMAT

See manual page **pscount(3)**.

SEE ALSO

edrcintro(1), **pscount(3)**

NOTES

-

BUGS

-

AUTHOR

pscount.kp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2012 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

psjoin – concatenate PostScript files

SYNOPSIS

edrc/bin/psjoin [**-h**]

psjoin [**-a**] [**-s**] [**-p**] *files...*

AVAILABILITY

WA2L/edrc

DESCRIPTION

join concatenates several PostScript files and generate a single PostScript document. The output, concatenated PostScript document, will be written to the standard output.

The input PostScript files must comply with the DSC (Document Structuring Convention). **psjoin** can fail to work depends to the input PostScript file or combination of the input PostScript files.

OPTIONS

- h** usage message.
- a** Align first page of each documents to odd page, by inserting extra blank page after odd-paged documents - maybe useful when concatenating two-sided documents.
- s** Try to close unclosed save operators in the input files. This option may be useful when input PostScript files have save operators which doesn't have corresponding restore operators, and the joined PostScript file causes "limitcheck" PostScript error due to too deeply nested save operators.
- p** Force insert corresponding PostScript prolog/trailer codes into all pages. Normally, to reduce the size of the output file, **psjoin** try not to insert largest prolog/trailer codes repeatedly.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 unknown option.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), <http://homepage3.nifty.com/tsato/tools/psjoin.html>

NOTES

This man page has been written based on the HTML page <http://homepage3.nifty.com/tsato/tools/psjoin.html>. See there for additional excellent PostScript tools.

BUGS

-

AUTHOR

psjoin has been written by Tom Sato, <VEF00200@nifty.ne.jp>, and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pslist – list selected processes

SYNOPSIS

edrc/bin/pslist [**-h**]

pslist [*select_opts*] [**-l** | **-H**]

pslist [*select_opts*] [**-H**] [**-o** *output_format*]

select_opts ::= [**-u** *user*] [**-g** *group*] [**-p** *pid*] [**-P** *ppid*] [**-t** *tty*] [**-c** "*command*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

list selected processes in a cross operating system compatible format.

If the long (**-l**) or the output format option (**-o** *fmt*) is not used, only a list of the process-ids (**PID**) of the selected processes is printed. This might be useful when a certain set of processes have to be killed in a script.

All *select_opts* options provided are used to select the corresponding column(s) on the process table without the need of taking care, that also other column(s) might have the identical entry.

Therefore the command to list all PIDs of user *jdoue*:

```
ps -ef | egrep -v egrep |\
egrep -e '^jdoue[^\ ]+.*' |\
awk '{ print $3 }'
```

can be replaced by:

```
pslist -u jdoue
```

All selection option arguments (**-u**, **-g**, **-p**, **-P**, **-t**, **-c**) can also be regular expressions. If the selection argument contains spaces, it has to be quoted.

When no selection option is used, the whole process table is printed.

OPTIONS

-h usage message.

-H print the process table column headers depending on the output. By default the header is not printed. The header goes to **stderr**, while the process list output goes to **stdout**.

-l long list output. When this option is used header and the columns of the process table as explained below are printed.

Therefore the **-l** option is equal to the command call: **pslist -H -o user,group,pid,ppid,TTY,command**.

Columns:

USER effective user name. This will be the textual user ID if it can be obtained, the decimal representation otherwise.

GROUP effective group ID of the process. This will be the textual group ID if it can be obtained, the decimal representation otherwise.

PID process ID number of the process.

PPID parent process ID.

TTY controlling tty (terminal).

COMMAND

command with all its arguments as a string. Modifications to the arguments may be shown. The output in this column may contain spaces.

-u *username*
username (USER) of the process to be selected.

-g *groupname*
groupname (GROUP) of the process to be selected.

-p *process-id*
process-id (PID) to be selected.

-P *parent-process-id*
parent-process-id (PPID) to be selected.

-t *Tty* tty of the process (TT) to be selected.

-c "*command*"
command including arguments (COMMAND) of the process to be selected. The *command* has to be quoted, if the command selection contains spaces.

-o *output_format*
define the **pslist** output format. The *output_format* is a comma separated list of columns of the process table to be printed. The field names are: *user*, *group*, *pid*, *ppid*, *TTY*, *command*. The output is separated by one tabulator.

To produce a nice looking output, use **print_list(3)** to format the output of **pslist**:

```
pslist -l 2>&1 | print_list "\t"
```

or

```
pslist -H -o user,pid,ppid,command 2>&1 | \  
print_list "\t"
```

ENVIRONMENT

-

EXIT STATUS

4 usage printed

0 no error

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ps(3)**, **pscount(3)**, **pstree(1)**, **regexintro(4)**, **xpid(1)**

NOTES

On Sun Solaris, only the processes related to the global zone are selected, therefore the processes of running zones on a global zone are not selected when **pslist** is started on the global zone. On zones the processes of the zone are listed.

Parts of the process table columns description were extracted from the **ps(1)** man page for procps version 2.0.7, written by Michael K. Johnson <johnsonm@redhat.com> that is distributed with RedHat Linux 7.2.

BUGS

pslist was successfully tested on: HPUX 11.11, HPUX 11.31, RedHat Linux 7.2, Suse Linux, Solaris 10.

AUTHOR

pslist was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ps2pdf – Convert PostScript to PDF using Ghostscript

SYNOPSIS

ps2pdf [options...] {input.[e]ps|-} [output.pdf|-]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **ps2pdf** command is a work-alike for nearly all the functionality (but not the user interface) of Adobe's Acrobat(TM) Distiller(TM) product: they convert PostScript files to Portable Document Format (PDF) files.

If the *output* filename is not specified, the output is placed in a file of the same name with a **'pdf'** extension in the current working directory. Either the *input* filename or the *output* filename can be **'-'** to request reading from **stdin** or writing to **stdout**, respectively, when used as a filter.

- **ps2pdf** per se currently produces PDF 1.4 output. However, this may change in the future. If you care about the compatibility level of the **output**, use the **-dCompatibilityLevel=1.x** switch in the command line.

There are some limitations in **ps2pdf**'s conversion. See the HTML documentation for more information. A large number of Adobe Distiller(TM) parameters which can be used to control the conversion are also documented there, including instructions for generating PDF/X and PDF/A documents.

OPTIONS

The **ps2pdf** command uses the same options as **gs(3)**.

EXAMPLES

Converting a *figure.ps* to *figure.pdf*:

```
ps2pdf figure.ps
```

A conversion with more specifics:

```
ps2pdf -dPDFSETTINGS=/prepress figure.ps proof.pdf
```

Converting as part of a pipe:

```
make_report.pl -t ps | ps2pdf -dCompatibilityLevel=1.3 - - | lpr
```

SEE ALSO

edrcintro(1), **gs(3)**, <https://ghostscript.readthedocs.io/en/latest/Use.html>, <https://ghostscript.readthedocs.io/en/latest/Use.html#convert-a-postscript-document-to-pdf>, <https://ghostscript.com/>

BUGS

See <http://bugs.ghostscript.com/> and the Usenet news group comp.lang.postscript.

VERSION

This document was last revised for Ghostscript version 10.00.

AUTHOR

Artifex Software, Inc. are the primary maintainers of Ghostscript. This manpage by George Ferguson.

This manpage of **ps2pdf** has been adapted from the original to reflect the integration into WA2L/edrc by Christian Walther. Send suggestions and bug reports regarding the integration to wa2l@users.sourceforge.net.

NAME

pstree – print a process tree

SYNOPSIS

edrc/bin/pstree [**-h**]

pstree [**-p** | **-r**] [*pid* | *processname*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print a tree of processes and the dependent sub-processes. Beside the process-ids, the process command is printed.

If no starting process-id (*pid*) is specified, the process tree is started beginning by the root process-id, else the tree starts with the specified *pid*.

OPTIONS

-h usage message.

-p print a list of the process-ids participating on the selected process tree.

-r print a list of the process-ids participating on the selected process tree in reverse order. This output can be used to kill processes of a process tree from leaf-to-root.

pid process-id of the process from which the process tree should start.

processname
processname from where to start the process tree.

The processname is the first process-id returned by the **pslist -c "processname"** command. Therefore the *processname* is a regular expression.

ENVIRONMENT

-

EXIT STATUS

| | |
|-----------|---|
| 0 | no error. |
| 1 | error. |
| 2 | operating system not supported. See osid(3) if you get this error. |
| 4 | usage printed. |
| 5 | the pstree command was aborted (pressing Ctrl+C). |
| 11 | temporary directory could not be claimed in /tmp/ , check the system if this error occurs, it is a sign of system intrusion. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **pid(1)**, **ps(1)**, **pscount(3)**, **pslist(3)**

NOTES

-

BUGS

-

AUTHOR

pstree was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

psup – start edrc with a special Production Support configuration

SYNOPSIS

edrc/bin/psup [**-h** | **-V**]

psup [**-s**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Short start of **edrc** with an other configuration which points to an own script tree. Internally **psup** calls **edrc -c edrc.psup.cfg -n PSUP_@ID@** .

psup stands for "Production SUPport".

The session name (as shown in the menu) is automatically set to **PSUP_<id>** . Where **<id>** is the process id of the started **edrc** instance if not already a session with the same name exists, if so the **<id>** is set to a random number.

If additional short starts are needed, create a symlink from the new short start command to **sat** (see section **EXAMPLES**).

OPTIONS

-h usage message.

-V print version and patch level of **edrc** For an explanation of the release numbering system see **edrcrevision(1)**.

-s silent startup. Startup without showing the EDRC banner.

-t no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

ENVIRONMENT

-

EXIT STATUS

see **edrc(1m)**.

FILES

etc/edrc.psup.cfg

configuration file of **psup**, see **edrc.cfg(4)** for more information.

Other files see section **FILES** in **edrc(1m)**.

EXAMPLES**1) create a new short start**

This creates a new short start command **new_shortstart** that will load the configuration file **edrc.new_shortstart.cfg**. The session name will automatically be set to **NEW_SHORTSTART_<id>** .

```
[ /root ]
[ root@rh7mzv7t001 ] [bash]: cd ~edrc/bin

[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ln -s sat new_shortstart
```

SEE ALSO

sat(1), **edrc(1m)**, **edrc.cfg(4)**, **edrcintro(1)**, **edrcrevision(1)**

NOTES

-

BUGS

-

AUTHOR

psup was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

purgetemp – purge temporary directories created by maketemp

SYNOPSIS

edrc/lib/purgetemp [**-h**]

purgetemp [**-s** *scriptname*] [**-q**] **-d** *directory*

AVAILABILITY

WA2L/edrc

DESCRIPTION

purge (=remove) all temporary directories that have the state **IS_ACTIVE=False** starting at a given *directory* (as listed by **listtemp -d directory**) created by **maketemp**(3).

Directories not created by **maketemp** will not be removed by the **purgetemp** command.

Use the **listtemp**(3) command to list the state of temporary directories created by **maketemp**.

OPTIONS

-h usage message.

-s *scriptname*
 scriptname in message output of **purgetemp**.

-q quiet, no message output.

-d *directory*
 temporary base directory as specified in the **listtemp** command.

See section **EXAMPLES** for additional information.

ENVIRONMENT

-

EXIT STATUS

0 no error.

- 3 specified temporary base directory does not exist.
- 4 usage message listed.

FILES

<directory>/
temporary base directory to be purged.

<directory>/<tempdir>/maketemp.flagfile
flagfile containing the state information about the command that created the temporary directory. This file is created by the **maketemp**(3) command while creating the temporary directory and read by **purgetemp**(3) and **listtemp**(3).

EXAMPLES

1) purge tempdirs created in '/tmp':

```
[ /root ]
[ root@acme001 ][-sh]: maketemp -d /tmp/.example
/tmp/.example.3430.2Hpt0D/

[ /root ]
[ root@acme001 ][-sh]: purgetemp -d /tmp
```

SEE ALSO

edrcintro(1), **listtemp**(3), **maketemp**(3), **removetemp**(3)

NOTES

-

BUGS

if the tempdir is created in a temporary directory location where a regular file cleanup is configured, **purgetemp** will not be able to purge the directory specified in the **-d** option, due to the fact that the **.maketemp.flagfile** might be removed by the cleanup function.

Cleanup **crontab** entry that would cause the behaviour described:

```
#
# delete all files in /tmp which are older than one week
#
00 1 * * * find /tmp \( -mtime +7 -a -type f \) -exec /bin/rm {} \;
```

AUTHOR

purgetemp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pwcrypt – encrypt a password from stdin

SYNOPSIS

edrc/bin/pwcrypt

AVAILABILITY

WA2L/edrc

DESCRIPTION

encrypt a plain password.

The plain password can be entered interactively or can be passed via pipe or input redirection.

The crypted password can be used in **/etc/passwd** of SunOS, Solaris, HP-11 (untrusted), HP-11ia (untrusted) and Linux.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **osid**(3)

NOTES

-

BUGS

-

AUTHOR

pwcrypt was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pwsafe – query the password safe for a password

SYNOPSIS

edrc/lib/pwsafe [**-h**]

pwsafe [**-c** *config_file*] *user@system*

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **pwsafe**(3) command provides the possibility to access the password safe maintained by the contributed **edrc** command **contrib.pwsafe**(1m) in read only mode from outside of an **edrc** session.

OPTIONS

-h print usage message.

-c *config_file*

edrc configuration file, see **edrc.cfg**(4) for more information.

This is to locate the password safe file within the related recovery tree.

user@system

the *user* on the related *system* or application to be queried.

ENVIRONMENT**\$PWSAFE_MASTERPASSWORD**

This environment variable must be set to the master password of the password safe to be able to query an encrypted password safe.

Be careful on how to handle the master password:

When writing a script accessing the **pwsafe** never supply the master password thru a command line option due to the fact that any user on the system would be able to see the password using the **ps -ef** command.

Therefore only use environment variables to pass the password around.

In addition the password shall not be saved in clear text.

EXIT STATUS

- | | |
|-----------|--|
| 0 | no error. |
| 1 | wrong master password. |
| 2 | operating system not supported. |
| 4 | usage message printed. |
| 5 | pwsafe aborted. |
| 6 | the given <i>config_file</i> does not exist in the edrc/etc/ directory. |
| 7 | the SCRIPTS_BASEDIR as defined in <i>config_file</i> does not exist. |
| 8 | permission denied to read password safe file. |
| 9 | CRYPT_MODE as defined for contrib.pwsafe(1m) is not supported on the system. |
| 11 | temporary directory could not be claimed or created in /var/tmp . Check the system temporary directory /var/tmp if you get this error, it is an indicator of system intrusion. |
| 99 | the CRYPT_MODE as defined for contrib.pwsafe(1m) is not implemented, yet. |

FILES

- pwsafe.dat**
password safe file in a recovery script tree referenced by the *config_file*.
- edrc/etc/contrib.edrc.pwsafe.cfg**
optional configuration file of the contributed command **contrib.pwsafe(1m)**.

EXAMPLES**1) using pwsafe in a script**

A simple script sniplet to show how to use the **pwsafe** usage:

```
#!/bin/ksh

:
:
PWSAFE_MASTERPASSWORD=enigma
export PWSAFE_MASTERPASSWORD

appadm_password=~edrc/lib/pwsafe appadmin@WEBAPP `
appsys_password=~edrc/lib/pwsafe system@acme007 `
```

:
:

SEE ALSO

edrcintro(1), **contrib.edrc(1m)**, **contrib.pwsafe(1m)**

NOTES

-

BUGS

-

AUTHOR

pwsafe was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pythonenv – print environment needed to start Python3 scripts

SYNOPSIS

edrc/lib/pythonenv [-n]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print the environment used by **pythonenv** to access the Python3 packages/modules bundled with WA2L/edrc.

To set the environment prior to the execution of **python3**, invoke:

```
eval `pythonenv`
```

This command is **only** to be used to prepare the environment to execute Python3 scripts in "Recovery Script" trees.

When writing Python3 scripts for WA2L/edrc, do **not** use **pythonenv**, start the script thru the **.python_wrapper**. See **python_wrapper**(1) for information about integrating a Python3 script into WA2L/edrc.

OPTIONS

-n no **export** *VARIABLE* ... output.

ENVIRONMENT**\$PYTHONPATH**

this variable holds Python3 package/module locations additional to the standard applying to an installation. This variable will be expanded with the bundled Python3 module locations (in the Python3 virtual environment **venv**) when executing the **pythonenv** command.

\$VIRTUAL_ENV

path of the Python3 virtual environment (**venv**) for the related operating system id and Python3 version (.../**\$OSID**/*<major>.<minor>/*) bundled with WA2L/edrc.

\$PATH

command search path. This variable will be expanded with the path where the **python3** interpreter was found on the system when executing the **pythonenv** command. See also **python_wrapper**(1) for more information.

EXIT STATUS

0 always

FILES

edrc/etc/python_wrapper.cfg
configuration file for **.python_wrapper** and **pythonenv**.

edrc/lib/python/pym/

location of all bundled Python3 packages/modules. The related directory in **edrc/lib/python/pym/<OSID>/<major>.<minor>/lib/python<major>.<minor>/site-packages/** is pre-pended to the **\$PYTHONPATH** environment variable when executing **pythonenv**.

Where **<major>** is the major version number (e.g. 3) and **<minor>** is the minor Python3 version (e.g. 6).

EXAMPLES**1) Start a Python3 script from a recovery script**

To start a Python3 script that uses a bundled Python3 module from a recovery script, set the Python3 environment using the 'eval **'pythonenv'**' command in the recovery script and then call the **python3** interpreter.

Recovery script:

```
#!/bin/ksh
#
# 1:ascript - A Recovery Script
#
# [00] 08.02.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x
:
:
eval `pythonenv`
python3 ./pythonscript
:
```

Python3 script:

```
#
# pythonscript - Python3 script using bundled complex module
#
# [00] 18.03.2009 CWa Initial Version
#
:
:
from complex import Complex
:
```

2) Start a Python3 script from a recovery script using the `_env` file mechanism

Create an `_env` file (using the `env edrc` command) and start the Python3 script from the recovery script.

`_env` file:

```
#
# _env - Environment settings for commands in /apps/eg
#
# [00] 18.03.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

eval `pythonenv`
:
```

Recovery script:

```
#!/bin/ksh
#
# 1:ascript - A Recovery Script
#
# [00] 08.02.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x
:
:
python3 ./pythonscript
:
```

Python3 script:

```
#
# pythonscript - Python3 script using bundled complex module
#
# [00] 18.03.2009 CWa Initial Version
#
:
:
from complex import Complex
:
```

3) Write a recovery script in Python3

Create an `_env` file (using the `env edrc` command) and write the recovery script in Python3.

`_env` file:

```
#
# _env - Environment settings for commands in /apps/eg
```

```
#
# [00] 18.03.2009 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

eval `pythonenv`
:
```

Python3 recovery script:

```
#!/usr/bin/env python3
#
# 1:ascrip - A Recovery Script
#
# [00] 08.02.2009 CWa Initial Version
#
:
:
from complex import Complex
:
```

Note the use of `#!/usr/bin/env python3` in the magic key to start the Python3 script. This ensures that the **python3** interpreter is found on the system, even when it is not installed in the default location you will normally specify. The path to the **python3** interpreter is added to the **\$PATH** using the **pythonenv** command.

4) Write a Python3 script outside of WA2L/edrc and profit from bundled Python3 modules

The startup method method below allows to start the **python3** interpreter from wherever it is installed and also enables to use all **python3** packages/modules bundled with WA2L/edrc using the **pythonenv** command to dynamically initialize the environment.

This method is similar to the use of the **.python_wrapper**, but the Python3 script can be placed somewhere of your liking outside of WA2L/edrc.

Set the permissions of the script to executable:

```
chmod +x pythonscript
```

Python3 script:

```
#!/bin/ksh
#
# pythonscript - Python3 script using bundled modules
#
# [00] 18.03.2009 CWa Initial Version
#

eval `~edrc/lib/pythonenv`

sed '1,/^exit \$\?/d' $0 | python3 - "$0" "$@"
exit $?
```

```
# python
#
import sys
from complex import Complex

sys.argv[0] = sys.argv.pop(1); __file__ = sys.argv[0]

def main():
    print("Hello World!\n");

    print("Arguments:", sys.argv[0:])

    x = Complex(2,3)
    y = Complex(4,5)
    print("x+y = {}".format(x+y))

    return(0)

if __name__ == "__main__": sys.exit(main())
```

5) Test the availability of a specific Python3 package/module

Set the needed **python3** environment:

```
[ / ]
[ root@acme007 ][*edrc*/bash]: eval `pythonenv`
```

Try to access the **python3** module:

```
[ / ]
[ root@acme007 ][*edrc*/bash]: python -c "import complex"
```

If an error message like

```
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'complex'
```

appears, the package/module cannot be found, if the package/module can be found there is no output.

6) List the available Python3 packages/modules

To list the installed Python3 packages, use the **pip3(1)** command.

To list the packages bundled in WA2L/edrc:

```
[ /opt/edrc/bin ]
[ root@acme007 ][*edrc*/bash]: pip list --local
```

To list all packages available (this includes the packages/modules bundled in WA2L/edrc):

```
[ /opt/edrc/bin ]  
[ root@acme007 ][*edrc*/bash]: pip list
```

7) Show detail information of an installed package

To list information about installed packages, the **pip3**(1) command is used.

The **pip show** [**--file**] *packagename* also displays the **Location** where the package is installed/read.

To show detail information of a package:

```
[ /opt/edrc/bin ]  
[ root@acme007 ][*edrc*/bash]: pip show complex
```

To show detail information and all installed files of a package:

```
[ /opt/edrc/bin ]  
[ root@acme007 ][*edrc*/bash]: pip show --file complex
```

SEE ALSO

edrcintro(1), **osid**(3), **pkgdir**(1m), **python3**(1), **pythonversion**(3), **pip**(1), **pip3**(1), **python_wrapper**(1), <https://pypi.org/>

NOTES

-

BUGS

-

AUTHOR

pythonev was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

pythonversion – print version (major.minor) of Python3 interpreter

SYNOPSIS

edrc/lib/pythonversion

AVAILABILITY

WA2L/edrc

DESCRIPTION

print Python3 version of **python3** interpreter found on the system.

See also **python_wrapper**(1) for additional information.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **apprevison**(3), **pythonenv**(3), **python_wrapper**(1), **revision**(3), **scriptrevision**(3)

NOTES

-

BUGS

-

AUTHOR

pythonversion was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

python_wrapper – wrap Python3 scripts to ensure python3 startup

SYNOPSIS

edrc/bin/pythonscript -> **.python_wrapper**

edrc/lib/pythonscript -> **.python_wrapper**

AVAILABILITY

WA2L/edrc

DESCRIPTION

Wrap Python3 scripts to avoid using `#!/usr/bin/python3` or similar path names in the magic key. This enhances the portability of WA2L/edrc commands written in Python3 to systems where **python3** is not installed in the default install locations.

python3 has to be installed in `/usr/bin/`, `/bin/`, `/sbin/`, `/usr/local/bin/` or `/usr/contrib/bin/`.

Further **python3** locations can be configured in the optional config file **python_wrapper.cfg**.

The Python3 packages/modules bundled with WA2L/edrc that are installed in **lib/python/pym/<OS-Id>/<major>.<minor>/** are included in the Python3 path (**PYTHONPATH**) and the Python3 virtual environment **venv** (**VIRTUAL_ENV**).

To start a new command thru the **.python_wrapper**, follow the following steps:

- 1.) create a symlink to the new command in the **edrc/bin/** or **edrc/lib/** directory depending on your needs:

```
[ /opt/edrc/bin ]
[ root@acme001 ][*edrc*/bash]: ln -s .python_wrapper new_cmd
```

- 2.) install the Python3 script in the **edrc/lib/python/** directory:

```
[ /opt/edrc/lib/python ]
[ root@acme001 ][*edrc*/bash]: vi new_cmd

[ /opt/edrc/lib/python ]
[ root@acme001 ][*edrc*/bash]: chmod 644 new_cmd
```

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

- 101** shell (**python3**) not found. If **python3** is installed on the system and you get this error, add the **python3** location in the configuration file **python_wrapper.cfg**.
- 102** the Python3 script to be started that should be located in **lib/python/** does not exist.
- 103** user calling the command has no permission to access/execute the called *pythonscript*.
- 107** the **.python_wrapper** was called directly.

FILES

- etc/python_wrapper.cfg**
configuration file for the **.python_wrapper** command.
- lib/python/**
location of the wrapped Python3 scripts. This files should have the file permissions *644* to show, that those scripts should not be started directly.
- lib/python/pym/<OS-Id>/<python-revision>/**
location of the Python3 packages/modules bundled with WA2L/edrc.

EXAMPLES

-

SEE ALSO

edrcintro(1), **binprobe(1m)**, **cmdlist(1m)**, **daemon_wrapper(1)**, **java_wrapper(1)**, **ksh_wrapper(1)**, **lua_wrapper(1)**, **perl_wrapper(1)**, **ln(1)**, **ld(1)**, **osid(3)**, **os_wrapper(1)**, **pip(1)**, **pip3(1)**, **python3(1)**, **pythonenv(3)**, **pythonversion(3)**, **python_wrapper.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

python_wrapper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

python_wrapper.cfg – configuration file for .python_wrapper

SYNOPSIS

edrc/etc/python_wrapper.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **.python_wrapper** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SEARCH_PATH**

Colon separated search path where to search for the **python3** interpreter.

Example: SEARCH_PATH="/usr/bin:/bin:/sbin:/usr/local/bin:/usr/contrib/bin"

Default: SEARCH_PATH="/usr/bin:/bin:/sbin:/usr/local/bin:/usr/contrib/bin"

SEE ALSO

edrcintro(1), python_wrapper(1)

NOTES

-

BUGS

-

AUTHOR

python_wrapper.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

random – return a random number or list item

SYNOPSIS

edrc/lib/random *number*

random *item1 item2 item3 ...*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return a random number or a random item of a specified item list.

OPTIONS

number maximum number to be randomly returned.

item1 item2 ...
item list. **random** will randomly return an item of the specified list.

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1)

NOTES

-

BUGS

-

AUTHOR

random was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rcat – display (cat) remote- or local file(s)

SYNOPSIS

edrc/bin/rcat [**-h**]

rcat [**-l** *localuser*] **[[***user@***]***host***:***file* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

display (**cat**) remote- or local file(s).

If the given *file* is **zipped** (**.zip**), **compressed** (**.Z**), **gzipped** (**.gz**), **bzip2ed** (**.bz**, **.bz2**, **.bzip2**) or **xzed** (**.xz**), the related uncompressed content is displayed.

OPTIONS

-h usage message.

[[*user@***]***host***:***file* ...
list of files to be displayed.

-l *localuser*
use the local user *localuser* to initiate the connection.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 error(s) while processing the given file(s) occurred.

4 usage printed.

FILES**etc/ssh-exec.cfg**configuration file of **ssh-exec**.The **rcat** command reads the **START_USER** setting only.**EXAMPLES**

-

SEE ALSO**edrcintro(1)**, **edrcsetup(1m)**, **bzip2(1)**, **comm(1)**, **compress(1)**, **diff(1)**, **funzip(1)**, **gzip(1)**, **rdiff(1)**, **rcomm(1)**, **ssh-exec(1)**, **ssh-exec.cfg(4)**, **unzip(1)**, **xz(1)**, **zip(1)****NOTES****rcat** uses **ssh-exec(1)** internally to connect to remote systems.**BUGS**

-

AUTHOR**rcat** was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.**COPYRIGHT**

Copyright © 2021 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rcmd – run command(s) on host(s)

SYNOPSIS

edrc/bin/rcmd [**-h**]

rcmd [**-n**][**-q**][**-r** [*num*]][**-t** *host_1,host_2,host_n*][**-c** *command*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **rcmd** command is used to run a command or a set of commands on a set of hosts. You can specify the command(s) on the command line, if you don't you are prompted for it.

The interactive query of hosts suggests as default the hosts as defined in the **EXEC_HOSTLIST** setting in the **rcmd.cfg** file. Furthermore it is possible to enter hostgroups (*@HOSTGROUP*), as known from the **hostlist** command. It is possible to specify multiple hostgroups or to mix hostgroups with hostnames. When using hostgroups the input default is expanded and the target prompt is repeated to give the user the opportunity to verify the target list and to correct it if needed.

All output as received from the commands started on the target hosts is printed to **stdout**, all **rcmd** state output and dialog is printed to **stderr** to enable the user to do further output processing of the received data, see also **-n** and **-q** options.

OPTIONS

-h usage message.

-n Do not indent the command output as received from the target host. This is useful if it is needed to do further output processing.

-q Quiet run, no state information is printed to **stderr**. To process the received output it is not imperative to use the **-q** option.

-r [*retries*]
number of retries to connect to the target host, if it was not possible to connect at the first try. This option can be used when using **rcmd** in unattended script runs. If *retries* is not set, the default as set in **CONNECTION_RETRIES** in the **rcmd.cfg** config file applies. If **-r** is not used in non-interactive script runs, this default applies, too. The interval of retries is eight seconds, if not specified differently in **RETRY_INTERVAL** in the **etc/rcmd.cfg** configuration file.

-t *hostlist* A comma separated list of target hosts. If the hostlist is not specified, an interactive query asks for it. The default hostlist for the interactive query is as defined in the **EXEC_HOSTLIST** setting in the **rcmd.cfg** config file.

See also **resolve_targetlist(3)** to see how to efficiently create a hostlist in scripts.

-c *command*

commands to be executed on the target nodes. If the commands are not specified on the command line a convenient interactive query allows to input the commands, even easy input editing is provided.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 2** operating system is not supported, yet. See **osid(3)** if you get this error.
- 4** usage printed.
- 5** the execution has been aborted using *Ctrl+C*.
- 6** the configuration file **rcmd.cfg** does not exist.
- 11** temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

- etc/rcmd.cfg**
configuration file of **rcmd**, see **rcmd.cfg(4)** for more information.
- var/log/rcmd.log**
default logfile of **rcmd**.
- etc/exrc** settings for the **vi** / **vim** editor.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **hostaliases(3)**, **hostlist(3)**, **remote_shell(3)**, **remote_copy(3)**, **rcmd.cfg(4)**, **resolve_targetlist(3)**, **ssh-exec(1)**, **syspoll(1)**

NOTES

-

BUGS

-

AUTHOR

rcmd was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rcmd.cfg – configuration file for rcmd

SYNOPSIS

edrc/etc/rcmd.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **rcmd** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **rcmd**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

DIST_HOSTLIST

Space separated list of hosts. To this hosts the files specified on the command line will be distributed.

Example: EXEC_HOSTLIST="hostlist"

Default: –

START_USER

User that is allowed to start **rcmd**.

Example: START_USER=eroot

Default: START_USER=root

EXEC_USER

User used to connect to the target hosts. This user has to be allowed to switch to root on the target system using the correct security settings. See **edrcsetup**(1m) for more information.

Example: EXEC_USER=edrc

Default: EXEC_USER=edrc

EXEC_MODE

Comma separated list of modes used to connect to the target hosts. The supported modes are: *rtools* which result in the use of **rcp** and **rsh** or *OpenSSH* which results in the use of **scp** and **ssh** for distribution. If a comma separated list is provided, the connection initiation is made in the sequence specified. A pseudo distribution mode is *default* which results in the use of the **EXEC_MODE** specified in the configuration files **remote_shell.cfg** and **remote_copy.cfg**. It is not allowed to specify *default* as part of a comma separated list.

Example: EXEC_MODE=rtools,OpenSSH

Default: EXEC_MODE=rtools

EXEC_SHELL

This is the shell the entered commands will be executed on the host. The shell has to be Bourne Shell compatible if you turn **SET_PATH** to *True*.

Example: EXEC_SHELL=/bin/ksh

Default: EXEC_SHELL=/bin/sh

CONNECTION_RETRIES

Number of connection retries if the connection to the target host fails. This setting applies if **-r** without specifying a number of retries is used or if **rcmd** is called non-interactive.

Example: CONNECTION_RETRIES=5

Default: CONNECTION_RETRIES=2

RETRY_INTERVAL

Seconds between retry attempts.

Example: RETRY_INTERVAL=10

Default: RETRY_INTERVAL=8

SET_PATH

The shell has to be Bourne Shell compatible if you turn **SET_PATH** to *True*. If you set **SET_PATH** to *True*, the contents of the local \$PATH variable will be appended to the \$PATH variable on the remote system.

Example: SET_PATH=True

Default: SET_PATH=False

SET_UMASK

If set to *True* the umask on the remote system will be set as it is set on the local system.

Example: SET_UMASK=True

Default: SET_UMASK=False

SUMMARY

If set to *True* at the end of the remote command execution a state summary is printed. If set to *False* the summary is not printed.

Example: SUMMARY=True

Default: SUMMARY=False

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **rcmd(1)**, **hostlist(3)**

NOTES

-

BUGS

-

AUTHOR

rcmd.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rcomm – compare (comm) remote- or local file(s)

SYNOPSIS

edrc/bin/rcomm [**-h**]

rcomm [*options*] [[*user@*]*host*:]*file1* [[*user@*]*host*:]*file2*

rcomm [*options*] [*user@*]*host*:*file1*

rcomm [*options*] [[*user@*]*host*:]*file1* [*user@*]*host*:

rcomm [*options*] [[*user@*]*host*:] [[*user@*]*host*:]*file2*

options ::= [**-l** *localuser*] [*diff_options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

compare (**comm**) remote- or local file(s).

When only specifying a remote file, it is compared with the related local file.

Example:

```
[ /opt/edrc/bin ]
[ root@acme-008 ] [*edrc*/bash]: rcomm acme-007:/etc/hosts
```

or

```
[ /opt/edrc/bin ]
[ root@acme-008 ] [*edrc*/bash]: rcomm acme-007: /etc/hosts
```

will compare the files **/etc/hosts** from the system acme-007 with the local **/etc/hosts** file on the acme-008 system.

The example above is therefore equal to:

```
[ /opt/edrc/bin ]
[ root@acme-008 ] [*edrc*/bash]: rcomm acme-007:/etc/hosts /etc/hosts
```

If the given file(s) are **gzipped** (**.gz**) or **bzip2ed** (**.bz2**, **.bz**), the related uncompressed content is compared.

OPTIONS

-h usage message.

[[user@]host:]file1
first remote- or local file.

[[user@]host:]file2
second remote- or local file.

When only specifying the first file, it is compared with the related local file.

-l localuser
use the local user *localuser* to initiate the connection. Default local user is **edrc**.
Use **-l \$USER** if you are not **root**.

comm_options
options of the **comm(1)** command.
See **comm(1)** for a description of the available **comm** options.

ENVIRONMENT

-

EXIT STATUS

0 no error.
1 error(s) while processing the given file(s) occurred.
4 usage printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **comm(1)**, **diff(1)**, **rcat(1)**, **rdiff(1)**, **ssh-exec(1)**, **ssh-exec.cfg(4)**

NOTES

rcomm uses **rcat**(1) internally to get the remote- and local files.

BUGS

-

AUTHOR

rcomm was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rdiff – show differences (diff) of remote- or local file(s)

SYNOPSIS

edrc/bin/rdiff [**-h**]

rdiff [*options*] [[*user@*]*host*:]*file1* [[*user@*]*host*:]*file2*

rdiff [*options*] [*user@*]*host*:*file1*

rdiff [*options*] [[*user@*]*host*:]*file1* [*user@*]*host*:

rdiff [*options*] [*user@*]*host*: [[*user@*]*host*:]*file2*

options ::= [**-l** *localuser*] [*diff_options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

show differences (**diff**) of remote- or local file(s).

When not specifying a second remote file, it is compared with the related local file or vice versa.

Example:

```
[ /opt/edrc/bin ]
[ root@acme-008 ][*edrc*/bash]: rdiff acme-007:/etc/hosts
```

or

```
[ /opt/edrc/bin ]
[ root@acme-008 ][*edrc*/bash]: rdiff acme-007: /etc/hosts
```

will compare the files **/etc/hosts** from the system acme-007 with the local **/etc/hosts** file on the acme-008 system.

The example above is therefore equal to:

```
[ /opt/edrc/bin ]
[ root@acme-008 ][*edrc*/bash]: rdiff acme-007:/etc/hosts /etc/hosts
```

If the given file(s) are **gzipped** (**.gz**) or **bzip2ed** (**.bz2**, **.bz**), the related uncompressed content is compared.

OPTIONS

-h usage message.

[[user@]host:]file1
first remote- or local file.

[[user@]host:]file2
second remote- or local file.

When only specifying the first file, it is compared with the related local file.

-l localuser
use the local user *localuser* to initiate the connection.

diff_options
options of the **diff**(1) command.
See **diff**(1) for a description of the available **diff** options.

ENVIRONMENT

-

EXIT STATUS

0 no error.
1 error(s) while processing the given file(s) occurred.
4 usage printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcsetup**(1m), **comm**(1), **diff**(1), **rcat**(1), **rcomm**(1), **ssh-exec**(1), **ssh-exec.cfg**(4)

NOTES

rdiff uses **rcat**(1) internally to get the remote- and local files.

BUGS

-

AUTHOR

rdiff was developed by Christian Walther. Send suggestions and bug reports to wa21@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

readline – read a long line from stdin

SYNOPSIS

edrc/lib/readline

AVAILABILITY

WA2L/edrc

DESCRIPTION

read long input lines from **stdin** and return the input to **stdout**.

The maximum length of the input is 10240 characters (10 kBytes), input longer than this maximum is ignored and truncated.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), input(3), read(2), readline(2)

NOTES

readline is used internally by the **input(3)**, command on Solaris and Linux.

BUGS

-

AUTHOR

readline was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

regexintro – introduction to regular expression usage

SYNOPSIS

regexintro, regex, regexp

AVAILABILITY

WA2L/edrc

DESCRIPTION

In computing, regular expressions provide a concise and flexible means for identifying strings of text of interest, such as particular characters, words, or patterns of characters.

Regular expressions (abbreviated as *regex* or *regexp*, with plural forms *regexes*, *regexps*, or *regexen*) are written in a formal language that can be interpreted by a regular expression processor, a program that either serves as a parser generator or examines text and identifies parts that match the provided specification.

BASIC CONCEPTS

A regular expression, often called a pattern, is an expression that describes a set of strings. They are usually used to give a concise description of a set, without having to list all elements.

For example, the set containing the three strings *Handel*, *Händel*, and *Haendel* can be described by the pattern **H(ä|ae?)ndel** (or alternatively, it is said that the pattern matches each of the three strings).

In most formalisms, if there is any regex that matches a particular set then there is an infinite number of such expressions. Most formalisms provide the following operations to construct regular expressions:

Alternation

A vertical bar separates alternatives. For example, **gray|grey** can match *gray* or *grey*.

Grouping Parentheses are used to define the scope and precedence of the operators (among other uses). For example, **gray|grey** and **gr(a|e)y** are equivalent patterns which both describe the set of *gray* and *grey*.

Quantification

A quantifier after a token (such as a character) or group specifies how often that preceding element is allowed to occur. The most common quantifiers are **?**, *****, and **+**.

? The question mark indicates there is zero or one of the preceding element. For example, **colou?r** matches both *color* and *colour*.

***** The asterisk indicates there are zero or more of the preceding element. For example, **ab*c** matches *ac*, *abc*, *abbc*, *abbbc*, and so on.

- + The plus sign indicates that there is one or more of the preceding element. For example, **ab+c** matches **abc**, **abbc**, **abbbc**, and so on, but not *ac*.

These constructions can be combined to form arbitrarily complex expressions, much like one can construct arithmetical expressions from numbers and the operations **+**, **.** and *****. For example, **H(ae?[ä]ndel** and **H(a|ae|ä)ndel** are both valid patterns which match the same strings as the earlier example, **H(ä|ae?)ndel**.

The precise syntax for regular expressions varies among tools and with context; more detail is given in the Syntax section.

SYNTAX

POSIX BASIC REGULAR EXPRESSIONS

Traditional Unix regular expression syntax followed common conventions but often differed from tool to tool.

The IEEE POSIX Basic Regular Expressions (BRE) standard (released alongside an alternative flavor called Extended Regular Expressions or ERE) was designed mostly for backward compatibility with the traditional syntax but provided a common standard which has since been adopted as the default syntax of many Unix regular expression tools, though there is often some variation or additional features.

Many such tools also provide support for ERE syntax with command line arguments.

In the BRE syntax, most characters are treated as literals - they match only themselves (i.e., **a** matches *a*). The exceptions, listed below, are called meta characters or meta sequences.

- .** Matches any single character (many applications exclude newlines, and exactly which characters are considered newlines is flavor, character encoding, and platform specific, but it is safe to assume that the line feed character is included). Within POSIX bracket expressions, the dot character matches a literal dot. For example, **a.c** matches *abc*, etc., but **[a.c]** **matches only** *a*, *.*, or *c*.
- []** A bracket expression. Matches a single character that is contained within the brackets. For example, **[abc]** matches *a*, *b*, or *c*. **[a-z]** specifies a range which matches any lowercase letter from *a* to *z*. These forms can be mixed: **[abcx-z]** matches *a*, *b*, *c*, *x*, *y* and *z*, as does **[a-cx-z]**.

The **-** character is treated as a literal character if it is the last or the first character within the brackets, or if it is escaped with a backslash: **[abc-]**, **[-abc]** or **[a-bc]**.
- [^]** Matches a single character that is not contained within the brackets. For example, **[^abc]** matches any character other than *a*, *b* or *c*. **[^a-z]** matches any single character that is not a lowercase letter from *a* to *z*. As above, literal characters and ranges can be mixed.
- ^** Matches the starting position within the string. In line-based tools, it matches the starting position of any line.
- \$** Matches the ending position of the string or the position just before a string-ending newline. In line-based tools, it matches the ending position of any line.

- *** Matches the preceding element zero or more times. For example, **ab*c** matches *ac*, *abc*, *abbbc* etc. **[xyz]*** matches *x*, *y*, *z*, *zx*, *zyx*, *xyzy* and so on. **\(ab\)*** matches *ab*, *abab*, *ababab* and so on.
- \{m,n\}** Matches the preceding element at least *m* and not more than *n* times. For example, **a\{3,5\}** matches only *aaa*, *aaaa* and *aaaaa*. This is not found in a few, older instances of regular expressions. For compatibility reasons, this construct should be avoided.

POSIX EXTENDED REGULAR EXPRESSIONS

The meaning of meta characters escaped with a backslash is reversed for some characters in the POSIX Extended Regular Expression (ERE) syntax. With this syntax, a backslash causes the meta character to be treated as a literal character. Additionally, support is removed for `\n` back references and the following meta characters are added:

- ?** Matches the preceding element zero or one time. For example, **ba?** matches *b* or *ba*.
- +** Matches the preceding element one or more times. For example, **ba+** matches *ba*, *baa*, *baaa* and so on.
- |** The choice (aka alternation or set union) operator matches either the expression before or the expression after the operator. For example, **abc|def** matches *abc* or *def*.

POSIX CHARACTER CLASSES

Since many ranges of characters depend on the chosen locale setting (i.e., in some settings letters are organized as *abc...zABC...Z*, while in some others as *aAbBcC...zZ*), the POSIX standard defines some classes or categories of characters as shown in the following table. It is expected, that this constructs are less portable, then specifying expressions with the more basic constructs above. Therefore for compatibility reasons, it is recommended to avoid the following constructs.

- [:alnum:]** Alphanumeric characters.
- [:alpha:]** Alphabetic characters.
- [:blank:]** Space and tab.
- [:cntrl:]** Control characters.
- [:digit:]** Digits.
- [:graph:]** Visible characters.
- [:lower:]** Lowercase letters.
- [:print:]** Visible characters and spaces.

[punct:] Punctuation characters.

[space:] White-space characters.

[upper:] Uppercase letters.

[xdigit:] Hexadecimal digits.

POSIX character classes can only be used within bracket expressions. For example, **[[:upper:]ab]** matches the uppercase letters and lowercase *a* and *b*.

EXAMPLES

- 1) **.at** matches any three-character string ending with *at*, including *hat*, *cat* and *bat*.
- 2) **[hc]at** matches *hat* and *cat*.
- 3) **[^b]at** matches all strings matched by **.at** except *bat*.
- 4) **^[hc]at** matches *hat* and *cat*, but only at the beginning of the string or line.
- 5) **[hc]at\$** matches *hat* and *cat*, but only at the end of the string or line.
- 6) **[hc]+at** matches *hat*, *cat*, *hhat*, *chat*, *hcat*, *ccchat* and so on, but not *at*.
- 7) **[hc]*at** matches *hat*, *cat*, *hhat*, *chat*, *hcat*, *ccchat* and so on, and also *at*.
- 8) **[hc]?at** matches *hat*, *cat* and *at*.
- 9) **cat|dog** matches *cat* or *dog*.
- 10) **.*** matches any character.

SEE ALSO

edrcintro(1), **awk(1)**, **egrep(1)**, **filedist.block(4)**, **is_osid(3)**, **logcheckd.pattern(4)**, **server_environment.cfg(4)**, **sed1line(1)**, **tr(1)**

[AWK] The AWK Programming Language, October 1988, Aho Alfred V., Weinberger Peter J., Kernighan Brian W., ISBN 0-201-07981-X

[REX] Regular Expression, Wikipedia the Free Encyclopedia, 14.06.2008, Boldt Axel, File: http://en.wikipedia.org/wiki/Regular_expression

[SSP] Shellscrip Programmierung, Sun Service, Revision C21 February 1994, Sun Microsystems Inc., Sun Part No: 8xx-xxxx-xx

NOTES

This manpage is an extract of the Wikipedia page http://en.wikipedia.org/wiki/Regular_expression which has been written by Boldt Axel and many others. See the mentioned web page to view the complete regular expression description.

BUGS

-

AUTHOR

regexintro was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rel2abs – convert relative file path name to absolute

SYNOPSIS

edrc/lib/rel2abs *filepath*

AVAILABILITY

WA2L/edrc

DESCRIPTION

convert a relative filename to an absolute.

OPTIONS

filepath file (with or without path) to be converted to an absolute path name.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1)

NOTES

-

BUGS

-

AUTHOR

rel2abs was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

remote_copy – start a remote copy

SYNOPSIS

edrc/lib/remote_copy [**-h**]

remote_copy [**-c** *cfg_file*] [**-m** *mode*[{*,mode*}]] [**-q**] [**-v**] [**-N**] [**-s** *username*] [**-l** *username*] *file_1*
[*file_2* ...] *dest*

remote_copy [**-c** *cfg_file*] **-f -m** *mode* [**-q**] [**-v**] [**-N**] [**-s** *username*] [**-l** *username*] *file_1* [*file_2* ...]
dest

AVAILABILITY

WA2L/edrc

DESCRIPTION

wrapper for remote copies.

This command enables you to write scripts independent of the current security policy of your systems.

Internally **remote_copy** uses **rcp** or **scp** dependent of the configuration settings.

Furthermore this command provides a caching mechanism which remembers successful connection modes to speed up future connections.

OPTIONS

-h usage message.

-c *cfg_file* config file (default=**edrc/etc/remote_copy.cfg**).

-m *mode* connection mode (default=**rtools**): **rtools**: rsh, rcp; **OpenSSH**: ssh, scp; **default**: default from config file.

-v verbose output.

-N do not **'su'** to the user given in **-s** *username* prior to invoking the connection.

-s *username* user to switch to on the local machine.

- f** force connection mode.
- l *username*** user to log in as on the remote machine.
- q** quiet, no error message output.
- file_n*** source file.
- dest*** destination file or directory in the form: [*user_name*@]*hostname:pathname/filename* or [*user_name*@]*hostname:pathname/dirname*

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 2** operating system not supported.
- 3** remote system is not up.
- 4** usage displayed.
- 5** command aborted.
- 6** config file not found.
- 7** cannot log on to remote system.
- 1##** ## exit code of **rcp** or **scp** if not successful.

FILES

- edrc/etc/remote_copy.cfg**
configuration file.
- edrc/var/connection/security/**
connection security files (e.g. SSH keys).
- edrc/var/connection/cache/**
connection cache files.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc(1m)**, **edrc.cfg(4)**, **edrcsetup(1m)**, **remote_copy.cfg(4)**, **remote_shell(3)**, **ssh-keyadd(1m)**

NOTES

-

BUGS

-

AUTHOR

remote_copy was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`remote_shell` – start a remote shell

SYNOPSIS

`edrc/lib/remote_shell [-h]`

`remote_shell [-c cfg_file] [-m mode[{,mode}]] [-q] [-N] [-s username] [-l username] [-t] host [--] [command]`

`remote_shell [-c cfg_file] -f -m mode [-n] [-q] [-N] [-s username] [-l username] [-t] host [--] [command]`

AVAILABILITY

WA2L/edrc

DESCRIPTION

wrapper for remote shells.

This command enables you to write scripts independent of the current security policy of your systems.

Internally **remote_shell** uses **remsh**, **rsh** or **ssh** dependent of the configuration settings.

Furthermore this command provides a caching mechanism which remembers successful connection modes to speed up future connections.

OPTIONS

-h usage message.

-c *cfg_file* config file (default=**edrc/etc/remote_shell.cfg**).

-m *mode* connection mode (default=**rtools**): **rtools**: rsh, rcp; **OpenSSH**: ssh, scp; **default**: default from config file.

-n do not probe connection before executing the command.

-N do not 'su' to the user given in **-s** *username* prior to invoking the connection.

-s *username* user to switch to on the local machine.

-f force connection mode.

-l *username* user to log in as on the remote machine.

-q quiet, no error message output.

-t allocate a tty even if command is given.

host host to log on to.

-- mark end of **remote_shell** options.

command command to execute on the remote *host*.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 operating system not supported.

3 remote system is not up.

4 usage displayed.

5 command aborted.

6 config file not found.

7 cannot log on to remote system.

1## **##** exit code of **rsh** or **ssh** if not successful.

FILES

edrc/etc/remote_shell.cfg
configuration file.

edrc/var/connection/security/
connection security files (e.g. SSH keys).

edrc/var/connection/cache/
connection cache files.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc(1m)**, **edrc.cfg(4)**, **edrcsetup(1m)**, **remote_shell.cfg(4)**, **remote_copy(3)**, **ssh-exec(1)**, **ssh-keyadd(1m)**

NOTES

-

BUGS

-

AUTHOR

remote_shell was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

removetemp – remove temporary directory created by maketemp

SYNOPSIS

edrc/lib/removetemp [**-h**]

removetemp [**-s** *scriptname*] [**-q**] **-d** *tempdir*

AVAILABILITY

WA2L/edrc

DESCRIPTION

remove a temporary directory created by **maketemp**(3).

Directories not created by **maketemp** will not be removed by the **removetemp** command.

Use the **listtemp**(3) command to list the state of temporary directories created by **maketemp**.

OPTIONS

-h usage message.

-s *scriptname*
scriptname in message output of **removetemp**.

-q quiet, no message output.

-d *tempdir* temporary directory as returned by the **maketemp** command.

See section **EXAMPLES** for additional information.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 cannot remove specified temporary directory.

- 2 directory not removed because it was not created using the **maketemp**(3) command.
- 3 specified temporary directory does not exist.
- 4 usage message listed.
- 5 cannot access the specified temporary directory.

FILES

<tempdir>/
temporary directory to be removed.

<tempdir>/maketemp.flagfile
flagfile containing the state information about the command that created the temporary directory. This file is created by the **maketemp**(3) command while creating the temporary directory and read by **removetemp**(3) and **listtemp**(3).

EXAMPLES

1) remove a tempdir created in '/tmp':

```
[ /root ]
[ root@acme001 ] [-sh]: maketemp -d /tmp/.example
/tmp/.example.3430.2Hpt0D/

[ /root ]
[ root@acme001 ] [-sh]: removetemp -d /tmp/.example.3430.2Hpt0D/
```

2) create/use/remove a tempdir in '/tmp' in a script:

```
#!/bin/sh

Scriptname=`basename $0`
Tmp=`maketemp -s $Scriptname -d /tmp/.$Scriptname` || exit 11

:
:

cat /etc/passwd | sort > ${Tmp}sorted_passwd

:
:

removetemp -q -d ${Tmp}
```

SEE ALSO

edrcintro(1), **listtemp(3)**, **maketemp(3)**, **purgetemp(3)**,

NOTES

-

BUGS

if the tempdir is created in a temporary directory location where a regular file cleanup is configured, **removetemp** will not be able to remove the directory specified in the **-d** option, due to the fact that the **.maketemp.flagfile** might be removed by the cleanup function.

Cleanup **crontab** entry that would cause the behaviour described:

```
#
# delete all files in /tmp which are older than one week
#
00 1 * * * find /tmp \( -mtime +7 -a -type f \) -exec /bin/rm {} \;
```

AUTHOR

removetemp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

repeat – repeat a given command

SYNOPSIS

edrc/lib/repeat *count command options*

AVAILABILITY

WA2L/edrc

DESCRIPTION

the **repeat** command re-executes the single subsequent command for count number of times.

OPTIONS

count number of *command* executions.

command options
command to be executed.

ENVIRONMENT

-

EXIT STATUS

x exit code of the last execution of the *command*.

FILES

-

EXAMPLES

-

repeat(3)

Library Commands

repeat(3)

SEE ALSO

edrcintro(1)

NOTES

-

BUGS

-

AUTHOR

repeat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

resolve_targetlist – produce contrib.doc OUTPUT-EXAMPLE output from logfile

SYNOPSIS

edrc/lib/resolve_targetlist [**-h**]

resolve_targetlist [**-l**] { *host* | **@GROUP** }

AVAILABILITY

WA2L/edrc

DESCRIPTION

resolve a targetlist which may consist of a list of hosts and hostgroups and return the resolved unique list of hosts.

OPTIONS

-h usage message.

-l the list is comma separated (instead of space separated).

host host name (list).

@GROUP group name (list).

ENVIRONMENT

-

EXIT STATUS

4 usage printed.

0 no error.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **hostlist(3)**, **input_targets(3)**

NOTES

-

BUGS

-

AUTHOR

resolve_targetlist was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

revision – print revision of a file examined from the file header

SYNOPSIS

edrc/bin/revision [*selection*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Print revision of a selection of files in the current working directory examined from the file header.

This helps to get a quick overview of the revision and the last revision history entry of files located in a certain directory.

The file header must have the (very simple to remember) format:

[##] *DD.MM.YYYY INi Revision history text*

Binary files are not scanned for file header patterns.

OPTIONS

selection

selection as known from the **ls** command to select a subset of files from the current working directory.

EXAMPLES**1) Shellscript header**

```
#!/bin/sh
#
# busy - keep the line open
#
# [00] 28.02.2001 CWa   Initial Version
# [01] 02.03.2001 CWa   chg: comment text
#
```

2) Configuration file header

```
#
# /etc/hosts - internet host revisions
#
```

```
# [00] 28.02.2001 CWa    Initial Version
# [01] 08.08.2001 CWa    + timehost
#
#
127.0.0.1      localhost
192.168.75.1   rh7mzv7t001
102.168.75.9   timehost
```

3) Configuration file header

```
;
; /etc/samba/smb.conf - samba configuration file
;
; [00] 28.02.2001 CWa    Initial Version
; [01] 04.04.2001 CWa    chg: workgroup AGROUP -> MYGROUP
;
;
[global]
    workgroup = MYGROUP
    server string = %h
```

4) SQL script header

```
--
-- TopTen.sql - Reports lists DB statistics
--
-- [00] 31.1.2003 CBi    Initial
-- [01] 01.5.2003 CWa    fix: total_bytes -> total_kbytes
--
select
    a.tablespace_revision,
    a.bytes/1024 as total_kbytes,
```

5) Manpage file header

```
.\"
.\" revision.1 - manpage for revision
.\"
.\" [00] 30.11.2003 CWa Initial Version
.\" [01] 03.01.2003 CWa  ++
.\" [02] 05.01.2003 CWa  ++
.\"
```

SEE ALSO

edrcintro(1), name(1)

NOTES

-

BUGS

-

AUTHOR

revision was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rl – shortcut for the rlogin/ssh command

SYNOPSIS

edrc/bin/rl [**-h**]

rl [*options*] *host*

AVAILABILITY

WA2L/edrc

DESCRIPTION

shortcut for the **rlogin** command.

On newer (Linux) operating systems, the invocation of the **rlogin** command starts **ssh**.

See **rlogin**(1) and **ssh**(1) manpage for additional information.

OPTIONS

-h usage message.

options all **rlogin** respectively **ssh** options.

hostname hostname to connect to.

ENVIRONMENT

-

EXIT STATUS

x exit code of the **rlogin** respectively **ssh** command.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **rcmd(1)**, **remote_shell(3)**, **rlogin(1)**, **ssh(1)**, **ssh-exec(1)**

NOTES

-

BUGS

-

AUTHOR

rl was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rnano – Restricted mode for Nano's ANOther editor, an enhanced free Pico clone

SYNOPSIS

rnano [*OPTIONS*] [[*+LINE,COLUMN*] *FILE*]...

DESCRIPTION

This manual page briefly documents the **rnano** command.

nano is a small, free and friendly editor which aims to replace Pico, the default editor included in the non-free Pine package. Rather than just copying Pico's look and feel, **nano** also implements some missing (or disabled by default) features in Pico, such as "search and replace" and "go to line and column number".

rnano is a restricted version of **nano**, which only edits specific files and doesn't allow the user access to the filesystem or a command shell.

In restricted mode, **nano** will *not*:

- read or write to any file not specified on the command line;
- read any nanorc files;
- allow suspending;
- allow a file to be appended to, prepended to, or saved under a different name;
- use backup files or spell checking.

OPTIONS

+LINE,COLUMN

Places cursor at line number *LINE* and column number *COLUMN* (at least one of which must be specified) on startup, instead of the default of line 1, column 1.

-? Same as **-h** (**--help**).

-h (**--help**)

Show a summary of command line options and exit.

-V (**--version**)

Show the current version number and exit.

See the **nano(1)** manpage for the complete documentation of **nano**.

BUGS

Please send any comments or bug reports to **nano@nano-editor.org**.

The **nano** mailing list is available from **nano-devel@gnu.org**.

To subscribe, email to **nano-devel-request@gnu.org** with a subject of "subscribe".

HOMEPAGE

<http://www.nano-editor.org/>

AUTHOR

Chris Allegretta <chrisa@asty.org>, et al (see AUTHORS for details). This manual page was originally written by Thijs Kinkhorst <thijs@kinkhorst.com>, for the Debian system (but may be used by others).

NAME

role_option – resolve user role option

SYNOPSIS

edrc/bin/role_option [**-h**]

role_option [**-c** *cfg*] **-e** *environment* **-r** *role*[,*{role}*] **-o** *option*

role_option [**-c** *cfg*] **-A** [**-e** *environment*] [**-r** *role*[,*{role}*]] [**-o** *option*[,*{option}*]]

role_option [**-c** *cfg*] **-R** **-e** *environment*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the value of a certain option related to a role in an environment.

This function is used to support a role based user creation model based on role templates which can be defined in the relating configuration file **role_option.cfg**(4).

OPTIONS

-h usage message.

-e *environment*
server environment name.

-c *configfile*
configuration file (default=**/opt/edrc-DVLP/edrc/etc/role_option.cfg**).

-r *role* list of roles separated by comma.

-o *option* option name.

-R list all defined roles.

-A list all defined options.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---|
| 2 | operating system not supported. |
| 4 | usage printed. |
| 6 | config file <i>cfg</i> (default= edrc/etc/role_option.cfg) not found. |

FILES

edrc/etc/role_option.cfg
role definition file.

EXAMPLES

-

SEE ALSO

edrcintro(1), **role_option.cfg(4)**,

NOTES

-

BUGS

-

AUTHOR

role_option was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

role_option.cfg – role and option configuration for role_option

SYNOPSIS

edrc/etc/role_option.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the role/option definition for the **role_option** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

ENVIRONMENT;ROLE;WEIGHT;CUMULATE;OPTION;VALUE;

Rows starting with a # are considered as comments. Empty lines are allowed, too.

OPTIONS*ENVIRONMENT*

Environment where the **ROLE** is valid.

The *ENVIRONMENT* name should be chosen as the environments defined in **edrc/etc/server_environment.cfg** respectively returned by the **server_environment** command.

ROLE

user role.

WEIGHT

The definition of the highest *WEIGHT* when multiple options with the identical *OPTION* name exist for the selected *ROLE*(es) is returned.

CUMULATE

if multiple options with the identical *OPTION* name exist for the selected *ROLE*(s) all defined *VALUE*(s) are cumulated (concatenated) in a comma separated list when this column is set to **T**.

When set to **F** the *VALUE* of the *OPTION* with the highest *WEIGHT* is returned.

OPTION

option name.

VALUE

value of the *OPTION*.

EXAMPLES

```
#
# role_option.cfg - config file for role_option
#
# [00] 16.06.2005 CWa Initial Version
#
#
#
# Format:
#
# <environment>;<role>;<weight>;<cumulate>;<option>;<value>;
#
#####
##### EDRC Development Roles #####
#####

# sys / Sysadmin (EDRC Development)
#
EDRC_DVLP;sys;100;F;title;Sysadmin;
EDRC_DVLP;sys;100;T;owner;EDS;
EDRC_DVLP;sys;100;T;page;27ff;
EDRC_DVLP;sys;100;T;unix-user;y;
EDRC_DVLP;sys;100;F;win-user;y;
EDRC_DVLP;sys;100;F;unix-prmgrp;eds;
EDRC_DVLP;sys;100;T;unix-secgrp;psoft;
EDRC_DVLP;sys;100;T;unix-secgrp;cmmon;
EDRC_DVLP;sys;100;T;unix-secgrp;sics_if;
EDRC_DVLP;sys;100;T;unix-secgrp;tr_psoft;
EDRC_DVLP;sys;100;F;unix-homedir;y;
EDRC_DVLP;sys;100;T;unix-suto;ps84reco;
EDRC_DVLP;sys;100;T;unix-suto;ps84prod;
EDRC_DVLP;sys;100;F;unix-shell;/usr/bin/ksh;
EDRC_DVLP;sys;100;T;unix-suto;notfall;
EDRC_DVLP;sys;100;T;unix-suto;root;
EDRC_DVLP;sys;100;F;unix-cron;n;
EDRC_DVLP;sys;100;F;unix-at;n;
EDRC_DVLP;sys;100;F;unix-ftp;y;
EDRC_DVLP;sys;100;F;unix-pwaging;y;
EDRC_DVLP;sys;100;F;unix-userclass;edssys;
```

```

EDRC_DVLP;sys;100;T;unix-suto;sics_if;

# ops / Operating (EDRC Development)
#
EDRC_DVLP;ops;100;F;title;Operating;
EDRC_DVLP;ops;100;T;owner;EDS;
EDRC_DVLP;ops;100;T;page;27ff;
EDRC_DVLP;ops;100;F;unix-user;y;
EDRC_DVLP;ops;100;F;win-user;y;
EDRC_DVLP;ops;100;F;unix-prmgrp;eds;
EDRC_DVLP;ops;100;T;unix-secgrp;psoft;
EDRC_DVLP;ops;100;T;unix-secgrp;cmmon;
EDRC_DVLP;ops;100;T;unix-secgrp;sics_if;
EDRC_DVLP;ops;100;T;unix-secgrp;tr_psoft;
EDRC_DVLP;ops;100;F;unix-homedir;y;
EDRC_DVLP;ops;100;T;unix-suto;ps84reco;
EDRC_DVLP;ops;100;T;unix-suto;ps84prod;
EDRC_DVLP;ops;100;F;unix-shell;/usr/bin/ksh;
EDRC_DVLP;ops;100;T;unix-suto;notfall;
EDRC_DVLP;ops;100;T;unix-suto;root;
EDRC_DVLP;ops;100;F;unix-cron;n;
EDRC_DVLP;ops;100;F;unix-at;n;
EDRC_DVLP;ops;100;F;unix-ftp;y;
EDRC_DVLP;ops;100;F;unix-pwaging;y;
EDRC_DVLP;ops;100;F;unix-userclass;edssys;
EDRC_DVLP;ops;100;F;unix-suto;sics_if;

#####
##### User Default #####
#####

# default / User Defaults
#
MAINTENANCE;_default;100;F;title;User Default Settings;
MAINTENANCE;_default;100;T;owner;;
MAINTENANCE;_default;100;T;page;;
MAINTENANCE;_default;100;F;unix-user;y;
MAINTENANCE;_default;100;F;win-user;y;
MAINTENANCE;_default;100;F;unix-prmgrp;;
MAINTENANCE;_default;100;T;unix-secgrp;;
MAINTENANCE;_default;100;F;unix-homedir;n;
MAINTENANCE;_default;100;F;unix-shell;/usr/bin/ksh;
MAINTENANCE;_default;100;T;unix-sufrom;;
MAINTENANCE;_default;100;T;unix-suto;;
MAINTENANCE;_default;100;T;unix-ownedby;;
MAINTENANCE;_default;100;T;unix-ownerof;;
MAINTENANCE;_default;100;F;unix-cron;n;
MAINTENANCE;_default;100;F;unix-at;n;
MAINTENANCE;_default;100;F;unix-ftp;n;
MAINTENANCE;_default;100;F;unix-pwaging;y;
MAINTENANCE;_default;100;F;unix-noagelogin;n;
MAINTENANCE;_default;100;F;unix-userclass;;

TEST;_default;100;F;title;User Default Settings;

```

```

TEST;_default;100;T;owner;;
TEST;_default;100;T;page;;
TEST;_default;100;F;unix-user;y;
TEST;_default;100;F;win-user;y;
TEST;_default;100;F;unix-prmgrp;;
TEST;_default;100;T;unix-secgrp;;
TEST;_default;100;F;unix-homedir;n;
TEST;_default;100;F;unix-shell;/usr/bin/ksh;
TEST;_default;100;T;unix-sufrom;;
TEST;_default;100;T;unix-suto;;
TEST;_default;100;T;unix-ownedby;;
TEST;_default;100;T;unix-ownerof;;
TEST;_default;100;F;unix-cron;n;
TEST;_default;100;F;unix-at;n;
TEST;_default;100;F;unix-ftp;n;
TEST;_default;100;F;unix-pwaging;y;
TEST;_default;100;F;unix-noagelogin;n;
TEST;_default;100;F;unix-userclass;;

PREPRODUCTION;_default;100;F;title;User Default Settings;
PREPRODUCTION;_default;100;T;owner;;
PREPRODUCTION;_default;100;T;page;;
PREPRODUCTION;_default;100;F;unix-user;y;
PREPRODUCTION;_default;100;F;win-user;y;
PREPRODUCTION;_default;100;F;unix-prmgrp;;
PREPRODUCTION;_default;100;T;unix-secgrp;;
PREPRODUCTION;_default;100;F;unix-homedir;n;
PREPRODUCTION;_default;100;F;unix-shell;/usr/bin/ksh;
PREPRODUCTION;_default;100;T;unix-sufrom;;
PREPRODUCTION;_default;100;T;unix-suto;;
PREPRODUCTION;_default;100;T;unix-ownedby;;
PREPRODUCTION;_default;100;T;unix-ownerof;;
PREPRODUCTION;_default;100;F;unix-cron;n;
PREPRODUCTION;_default;100;F;unix-at;n;
PREPRODUCTION;_default;100;F;unix-ftp;n;
PREPRODUCTION;_default;100;F;unix-pwaging;y;
PREPRODUCTION;_default;100;F;unix-noagelogin;n;
PREPRODUCTION;_default;100;F;unix-userclass;;

PRODUCTION;_default;100;F;title;User Default Settings;
PRODUCTION;_default;100;T;owner;;
PRODUCTION;_default;100;T;page;;
PRODUCTION;_default;100;F;unix-user;y;
PRODUCTION;_default;100;F;win-user;y;
PRODUCTION;_default;100;F;unix-prmgrp;;
PRODUCTION;_default;100;T;unix-secgrp;;
PRODUCTION;_default;100;F;unix-homedir;n;
PRODUCTION;_default;100;F;unix-shell;/usr/bin/ksh;
PRODUCTION;_default;100;T;unix-sufrom;;
PRODUCTION;_default;100;T;unix-suto;;
PRODUCTION;_default;100;T;unix-ownedby;;
PRODUCTION;_default;100;T;unix-ownerof;;
PRODUCTION;_default;100;F;unix-cron;n;
PRODUCTION;_default;100;F;unix-at;n;

```

```

PRODUCTION;_default;100;F;unix-ftp;n;
PRODUCTION;_default;100;F;unix-pwaging;y;
PRODUCTION;_default;100;F;unix-noagelogin;n;
PRODUCTION;_default;100;F;unix-userclass;;

BACKUP;_default;100;F;title;User Default Settings;
BACKUP;_default;100;T;owner;;
BACKUP;_default;100;T;page;;
BACKUP;_default;100;F;unix-user;y;
BACKUP;_default;100;F;win-user;y;
BACKUP;_default;100;F;unix-prmgrp;;
BACKUP;_default;100;T;unix-secgrp;;
BACKUP;_default;100;F;unix-homedir;n;
BACKUP;_default;100;F;unix-shell;/usr/bin/ksh;
BACKUP;_default;100;T;unix-sufrom;;
BACKUP;_default;100;T;unix-suto;;
BACKUP;_default;100;T;unix-ownedby;;
BACKUP;_default;100;T;unix-ownerof;;
BACKUP;_default;100;F;unix-cron;n;
BACKUP;_default;100;F;unix-at;n;
BACKUP;_default;100;F;unix-ftp;n;
BACKUP;_default;100;F;unix-pwaging;y;
BACKUP;_default;100;F;unix-noagelogin;n;
BACKUP;_default;100;F;unix-userclass;;

```

SEE ALSO

edrcintro(1), **role_option(1)**

NOTES

-

BUGS

-

AUTHOR

role_option.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rosid – evaluate the system’s operating system id of a remote system

SYNOPSIS

edrc/lib/rosid [**-h**]

rosid -t *hostname*

rosid [**-e** | **-l** *localuser*] **-u** *user@hostname*

rosid [**-e** | **-l** *localuser*] [**-u** *user*] **-t** *hostname*

AVAILABILITY

WA2L/edrc

DESCRIPTION

rosid generates an unique identifier for each combination of OS release (**uname -r**), OS version (**uname -s**) and machine architecture (**uname -m**) of a remote system.

The **rosid(3)** command does not probe the operating system and also does not resolve an existing **/etc/sfi/OSID** file on the remote system; else the operating system id resolution is identical to the **osid(3)** command.

See **osid(3)** for more information.

OPTIONS

-h print **osid** usage.

-e use **EDRC_USER** as defined in **etc/ssh-exec.cfg** (default=**edrc**) as local user.

-l *localuser*
use the local user *localuser* to initiate the connection. Default local user is **edrc**.

-u *user* remote user to connect on the target system. Default remote user is **root**.

-t *hostname*
target system *hostname* to execute the *commands*.

-u *user@hostname*
remote *user* to connect on the target system *hostname*.

EXIT STATUS

- | | |
|----------|--|
| 0 | the operating system id could be resolved. |
| 1 | the operating system id could not be resolved. In this case rosid returns <i>unknown</i> as OS-id. If you receive this return value you should check the configuration file edrc/etc/osid.dat or edrc/etc/osid.dat.WA2L if the entry for the current operating system is missing. If two releases on an operating system are sufficient similar, they should be mapped to the same OS-id. |
| 2 | cannot connect to the remote system. |
| 4 | usage listed. |
| 5 | command aborted. |

EXAMPLES

See **osid**(3).

SEE ALSO

osid(3), **osid.cfg**(4), **osid.dat**(4), **sysconfig**(1), **edrcintro**(1)

FILES

edrc/etc/osid.cfg

configuration file for **osid** and **rosid**. See **osid.cfg**(4) for more information.

edrc/etc/osid.dat

mapfile of **uname -rs** or **uname -rsm** to OS-id. See **osid.dat**(4) for more information.

NOTES

The WA2L/edrc package does not need to be installed on the remote system.

BUGS

-

AUTHOR

rosid was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rsat – start sat on a remote system

SYNOPSIS

edrc/bin/rsat [**-h**]

rsat [**-c** *config_file*] [*sat_options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

start **sat** on a remote system without the need to manually log in to the remote system first.

The list of hosts provided can be specified in the configuration file **etc/rsat.cfg**.

OPTIONS

-h usage message.

-c *config_file*
config file of **rsat**, located in **edrc/etc/**.

Default is: **etc/rsat.cfg**.

sat_options
all options of **sat(1)**.

ENVIRONMENT

-

EXIT STATUS

2 operating system not supported.

4 usage printed.

5 command aborted.

6 *config_file* not found.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc(1m)**, **remote_shell(4)**, **rsat.cfg(4)**, **sat(1)**

NOTES

-

BUGS

-

AUTHOR

rsat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rsat.cfg – configuration file for rsat

SYNOPSIS

edrc/etc/rsat.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **rsat** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**CONNECTION_USER**

User used to connect to the remote host if the user starting rsat is root.

Example: CONNECTION_USER=edrc

Default: CONNECTION_USER=edrc

EDRC_USER

The home of this user is considered as the root of the EDRC installation on remote systems.

Example: EDRC_USER=edrc

Default: EDRC_USER=edrc

REMOTE_PATH

General Path on remote systems.

Example: REMOTE_PATH=\$PATH

Default: REMOTE_PATH=\$PATH

TARGET_HOSTS Space separated list of hosts. This hosts will appear in the host selection list.

Example: TARGET_HOSTS="hostlist -g @ALL"

Default: TARGET_HOSTS="hostlist"

SEE ALSO

edrcintro(1), **rsat(1)**,

NOTES

-

BUGS

-

AUTHOR

rsat.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rssh – remote direct or interactive ssh connect

SYNOPSIS

edrc/bin/rssh [**-h**]

rssh [**-l**]

rssh-user@host [**-h**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

connect to a remote system using **ssh**(1).

When invoking the **rssh** command the remote user and the remote host is queried interactively, where the last chosen user- and host-name is provided as default input.

When a symlink to the **rssh** command in the format **rssh-user@host** is created, the newly created command variant will connect directly to the *host* with the *user* and the remote system will query for the password.

The main purpose for this is to create a pseudo user (without password) and add the command variant **rssh-user@host** to the shell field of the **/etc/passwd** field to allow to use a local system to be used as a direct terminal to a remote system, as for example:

/etc/passwd:

```
:
mars:x:1291:1291:mars login terminal:/tmp:/home/fred/bin/rssh-fred@mars
:
```

/etc/shadow:

```
:
mars::18715:0:99999:7:::
:
```

This would allow you to enter *mars* as login on the local system that will connect to the remote *host* with name *mars* where the login password of the *user* with name *fred* will be queried.

OPTIONS

-h usage message.

-l list past connections for selection.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 the **rssh** command has been aborted.

4 usage displayed.

FILES

\$HOME/.rssh
remember *user* and *host* of last **rssh** connection of the invoking user.

/etc/passwd
user base.

/etc/shadow
user passwords.

EXAMPLES

1) create a new rssh connect variant

This creates a new **rssh** command variant on host *earth* to directly connect to the host *mars* using the remote user-name *fred*.

```
[ /home/fred ]  
[ fred@earth ] [bash]: ln -s ~edrc/bin/rssh ~fred/bin/rssh-fred@mars
```

or:

```
[ /home/fred ]  
[ fred@earth ] [bash]: symlink ~edrc/bin/rssh ~fred/bin/rssh-fred@mars
```

SEE ALSO

edrcintro(1), **ln(1)**, **remote_shell(3)**, **rl(1)**, **ssh(1)**, **ssh-exec(1)**, **symlink(1)**

NOTES

-

BUGS

-

AUTHOR

rssh was developed by Christian Walther. Send suggestions and bug reports to wa21@users.sourceforge.net

.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

rsync — a fast, versatile, remote (and local) file-copying tool

SYNOPSIS

Local: rsync [OPTION...] SRC... [DEST]

Access via remote shell:

Pull: rsync [OPTION...] [USER@]HOST:SRC... [DEST]

Push: rsync [OPTION...] SRC... [USER@]HOST:DEST

Access via rsync daemon:

Pull: rsync [OPTION...] [USER@]HOST::SRC... [DEST]

rsync [OPTION...] rsync://[USER@]HOST[:PORT]/SRC... [DEST]

Push: rsync [OPTION...] SRC... [USER@]HOST::DEST

rsync [OPTION...] SRC... rsync://[USER@]HOST[:PORT]/DEST

Usages with just one SRC arg and no DEST arg will list the source files instead of copying.

DESCRIPTION

Rsync is a fast and extraordinarily versatile file copying tool. It can copy locally, to/from another host over any remote shell, or to/from a remote rsync daemon. It offers a large number of options that control every aspect of its behavior and permit very flexible specification of the set of files to be copied. It is famous for its delta-transfer algorithm, which reduces the amount of data sent over the network by sending only the differences between the source files and the existing files in the destination. Rsync is widely used for back-ups and mirroring and as an improved copy command for everyday use.

Rsync finds files that need to be transferred using a “quick check” algorithm (by default) that looks for files that have changed in size or in last-modified time. Any changes in the other preserved attributes (as requested by options) are made on the destination file directly when the quick check indicates that the file’s data does not need to be updated.

Some of the additional features of rsync are:

- o support for copying links, devices, owners, groups, and permissions
- o exclude and exclude-from options similar to GNU tar
- o a CVS exclude mode for ignoring the same files that CVS would ignore
- o can use any transparent remote shell, including ssh or rsh
- o does not require super-user privileges
- o pipelining of file transfers to minimize latency costs
- o support for anonymous or authenticated rsync daemons (ideal for mirroring)

GENERAL

Rsync copies files either to or from a remote host, or locally on the current host (it does not support copying files between two remote hosts).

There are two different ways for rsync to contact a remote system: using a remote-shell program as the transport (such as ssh or rsh) or contacting an rsync daemon directly via TCP. The remote-shell transport is used whenever the source or destination path contains a single colon (:) separator after a host specification. Contacting an rsync daemon directly happens when the source or destination path contains a double colon (::) separator after a host specification, OR when an rsync:// URL is specified (see also the “USING RSYNC-DAEMON FEATURES VIA A REMOTE-SHELL CONNECTION” section for an exception to this latter rule).

As a special case, if a single source arg is specified without a destination, the files are listed in an output format similar to “ls -l”.

As expected, if neither the source or destination path specify a remote host, the copy occurs locally (see also the `--list-only` option).

SETUP

See the file README for installation instructions.

Once installed, you can use rsync to any machine that you can access via a remote shell (as well as some that you can access using the rsync daemon-mode protocol). For remote transfers, a modern rsync uses ssh for its communications, but it may have been configured to use a different remote shell by default, such as rsh or remsh.

You can also specify any remote shell you like, either by using the `-e` command line option, or by setting the RSYNC_RSH environment variable.

Note that rsync must be installed on both the source and destination machines.

USAGE

You use rsync in the same way you use rcp. You must specify a source and a destination, one of which may be remote.

Perhaps the best way to explain the syntax is with some examples:

```
rsync -t *.c foo:src/
```

This would transfer all files matching the pattern `*.c` from the current directory to the directory `src` on the machine `foo`. If any of the files already exist on the remote system then the rsync remote-update protocol is used to update the file by sending only the differences. See the tech report for details.

```
rsync -avz foo:src/bar /data/tmp
```

This would recursively transfer all files from the directory `src/bar` on the machine `foo` into the `/data/tmp/bar` directory on the local machine. The files are transferred in “archive” mode, which ensures that symbolic links, devices, attributes, permissions, ownerships, etc. are preserved in the transfer. Additionally, compression will be used to reduce the size of data portions of the transfer.

```
rsync -avz foo:src/bar/ /data/tmp
```

A trailing slash on the source changes this behavior to avoid creating an additional directory level at the destination. You can think of a trailing `/` on a source as meaning “copy the contents of this directory” as opposed to “copy the directory by name”, but in both cases the attributes of the containing directory are transferred to the containing directory on the destination. In other words, each of the following commands copies the files in the same way, including their setting of the attributes of `/dest/foo`:

```
rsync -av /src/foo /dest
rsync -av /src/foo/ /dest/foo
```

Note also that host and module references don’t require a trailing slash to copy the contents of the default directory. For example, both of these copy the remote directory’s contents into “`/dest`”:

```
rsync -av host: /dest
rsync -av host::module /dest
```

You can also use rsync in local-only mode, where both the source and destination don’t have a `‘:’` in the name. In this case it behaves like an improved copy command.

Finally, you can list all the (listable) modules available from a particular rsync daemon by leaving off the module name:

```
rsync somehost.mydomain.com::
```

And, if Service Location Protocol is available, the following will list the available rsync servers:

```
rsync rsync://
```

See the following section for even more usage details.

ADVANCED USAGE

The syntax for requesting multiple files from a remote host is done by specifying additional remote-host args in the same style as the first, or with the hostname omitted. For instance, all these work:

```
rsync -av host:file1 :file2 host:file{3,4} /dest/
rsync -av host::modname/file{1,2} host::modname/file3 /dest/
rsync -av host::modname/file1 ::modname/file{3,4}
```

Older versions of rsync required using quoted spaces in the SRC, like these examples:

```
rsync -av host:'dir1/file1 dir2/file2' /dest
rsync host::'modname/dir1/file1 modname/dir2/file2' /dest
```

This word-splitting still works (by default) in the latest rsync, but is not as easy to use as the first method.

If you need to transfer a filename that contains whitespace, you can either specify the **--protect-args (-s)** option, or you'll need to escape the whitespace in a way that the remote shell will understand. For instance:

```
rsync -av host:'file\ name\ with\ spaces' /dest
```

CONNECTING TO AN RSYNC DAEMON

It is also possible to use rsync without a remote shell as the transport. In this case you will directly connect to a remote rsync daemon, typically using TCP port 873. (This obviously requires the daemon to be running on the remote system, so refer to the **STARTING AN RSYNC DAEMON TO ACCEPT CONNECTIONS** section below for information on that.)

Using rsync in this way is the same as using it with a remote shell except that:

- o you either use a double colon :: instead of a single colon to separate the hostname from the path, or you use an rsync:// URL.
- o the first word of the “path” is actually a module name.
- o the remote daemon may print a message of the day when you connect.
- o if you specify no path name on the remote daemon then the list of accessible paths on the daemon will be shown.
- o if you specify no local destination then a listing of the specified files on the remote daemon is provided.
- o you must not specify the **--rsh (-e)** option.

An example that copies all the files in a remote module named “src”:

```
rsync -av host::src /dest
```

Some modules on the remote daemon may require authentication. If so, you will receive a password prompt when you connect. You can avoid the password prompt by setting the environment variable **RSYNC_PASSWORD** to the password you want to use or using the **--password-file** option. This may be useful when scripting rsync.

WARNING: On some systems environment variables are visible to all users. On those systems using **--password-file** is recommended.

You may establish the connection via a web proxy by setting the environment variable **RSYNC_PROXY** to

a hostname:port pair pointing to your web proxy. Note that your web proxy's configuration must support proxy connections to port 873.

You may also establish a daemon connection using a program as a proxy by setting the environment variable `RSYNC_CONNECT_PROG` to the commands you wish to run in place of making a direct socket connection. The string may contain the escape “%H” to represent the hostname specified in the `rsync` command (so use “%%” if you need a single “%” in your string). For example:

```
export RSYNC_CONNECT_PROG='ssh proxyhost nc %H 873'
rsync -av targethost1::module/src/ /dest/
rsync -av rsync:://targethost2/module/src/ /dest/
```

The command specified above uses `ssh` to run `nc` (netcat) on a proxyhost, which forwards all data to port 873 (the `rsync` daemon) on the targethost (%H).

USING RSYNC-DAEMON FEATURES VIA A REMOTE-SHELL CONNECTION

It is sometimes useful to use various features of an `rsync` daemon (such as named modules) without actually allowing any new socket connections into a system (other than what is already required to allow remote-shell access). `Rsync` supports connecting to a host using a remote shell and then spawning a single-use “daemon” server that expects to read its config file in the home dir of the remote user. This can be useful if you want to encrypt a daemon-style transfer's data, but since the daemon is started up fresh by the remote user, you may not be able to use features such as `chroot` or change the uid used by the daemon. (For another way to encrypt a daemon transfer, consider using `ssh` to tunnel a local port to a remote machine and configure a normal `rsync` daemon on that remote host to only allow connections from “localhost”.)

From the user's perspective, a daemon transfer via a remote-shell connection uses nearly the same command-line syntax as a normal `rsync-daemon` transfer, with the only exception being that you must explicitly set the remote shell program on the command-line with the `--rsh=COMMAND` option. (Setting the `RSYNC_RSH` in the environment will not turn on this functionality.) For example:

```
rsync -av --rsh=ssh host::module /dest
```

If you need to specify a different remote-shell user, keep in mind that the `user@` prefix in front of the host is specifying the `rsync-user` value (for a module that requires user-based authentication). This means that you must give the `-l user` option to `ssh` when specifying the remote-shell, as in this example that uses the short version of the `--rsh` option:

```
rsync -av -e "ssh -l ssh-user" rsync-user@host::module /dest
```

The “ssh-user” will be used at the `ssh` level; the “rsync-user” will be used to log-in to the “module”.

STARTING AN RSYNC DAEMON TO ACCEPT CONNECTIONS

In order to connect to an `rsync` daemon, the remote system needs to have a daemon already running (or it needs to have configured something like `inetd` to spawn an `rsync` daemon for incoming connections on a particular port). For full information on how to start a daemon that will handling incoming socket connections, see the **rsyncd.conf(5)** man page — that is the config file for the daemon, and it contains the full details for how to run the daemon (including stand-alone and `inetd` configurations).

If you're using one of the remote-shell transports for the transfer, there is no need to manually start an `rsync` daemon.

EXAMPLES

Here are some examples of how I use `rsync`.

To backup my wife's home directory, which consists of large MS Word files and mail folders, I use a cron job that runs

```
rsync -Cavz . arvidsjaur:backup
```

each night over a PPP connection to a duplicate directory on my machine “arvidsjaur”.

To synchronize my samba source trees I use the following Makefile targets:

```
get:
    rsync -avuzb --exclude '*' samba:samba/ .
put:
    rsync -Cavuzb . samba:samba/
sync: get put
```

this allows me to sync with a CVS directory at the other end of the connection. I then do CVS operations on the remote machine, which saves a lot of time as the remote CVS protocol isn't very efficient.

I mirror a directory between my “old” and “new” ftp sites with the command:

```
rsync -az -e ssh --delete ~ftp/pub/samba nimbus:~ftp/pub/tridge"
```

This is launched from cron every few hours.

OPTIONS SUMMARY

Here is a short summary of the options available in rsync. Please refer to the detailed description below for a complete description.

| | |
|---------------------|---|
| -v, --verbose | increase verbosity |
| -q, --quiet | suppress non-error messages |
| --no-motd | suppress daemon-mode MOTD (see caveat) |
| -c, --checksum | skip based on checksum, not mod-time & size |
| -a, --archive | archive mode; equals -rlptgoD (no -H, -A, -X) |
| --no-OPTION | turn off an implied OPTION (e.g. --no-D) |
| -r, --recursive | recurse into directories |
| -R, --relative | use relative path names |
| --no-implied-dirs | don't send implied dirs with --relative |
| -b, --backup | make backups (see --suffix & --backup-dir) |
| --backup-dir=DIR | make backups into hierarchy based in DIR |
| --suffix=SUFFIX | backup suffix (default ~ w/o --backup-dir) |
| -u, --update | skip files that are newer on the receiver |
| --inplace | update destination files in-place |
| --append | append data onto shorter files |
| --append-verify | --append w/old data in file checksum |
| -d, --dirs | transfer directories without recursing |
| -l, --links | copy symlinks as symlinks |
| -L, --copy-links | transform symlink into referent file/dir |
| --copy-unsafe-links | only "unsafe" symlinks are transformed |
| --safe-links | ignore symlinks that point outside the tree |
| -k, --copy-dirlinks | transform symlink to dir into referent dir |
| -K, --keep-dirlinks | treat symlinked dir on receiver as dir |
| -H, --hard-links | preserve hard links |
| -p, --perms | preserve permissions |
| -E, --executability | preserve executability |
| --chmod=CHMOD | affect file and/or directory permissions |
| -A, --acls | preserve ACLs (implies -p) |
| -X, --xattrs | preserve extended attributes |
| -o, --owner | preserve owner (super-user only) |
| -g, --group | preserve group |
| --devices | preserve device files (super-user only) |
| --specials | preserve special files |
| -D | same as --devices --specials |
| -t, --times | preserve modification times |


```

-O, --omit-dir-times    omit directories from --times
  --super               receiver attempts super-user activities
  --fake-super          store/recover privileged attrs using xattrs
-S, --sparse            handle sparse files efficiently
-n, --dry-run           perform a trial run with no changes made
-W, --whole-file        copy files whole (w/o delta-xfer algorithm)
-x, --one-file-system   don't cross filesystem boundaries
-B, --block-size=SIZE   force a fixed checksum block-size
-e, --rsh=COMMAND       specify the remote shell to use
  --rsync-path=PROGRAM  specify the rsync to run on remote machine
  --existing             skip creating new files on receiver
  --ignore-existing     skip updating files that exist on receiver
  --remove-source-files sender removes synchronized files (non-dir)
  --del                 an alias for --delete-during
  --delete              delete extraneous files from dest dirs
  --delete-before       receiver deletes before transfer (default)
  --delete-during       receiver deletes during xfer, not before
  --delete-delay        find deletions during, delete after
  --delete-after        receiver deletes after transfer, not before
  --delete-excluded     also delete excluded files from dest dirs
  --ignore-errors       delete even if there are I/O errors
  --force               force deletion of dirs even if not empty
  --max-delete=NUM      don't delete more than NUM files
  --max-size=SIZE       don't transfer any file larger than SIZE
  --min-size=SIZE       don't transfer any file smaller than SIZE
  --partial             keep partially transferred files
  --partial-dir=DIR     put a partially transferred file into DIR
  --delay-updates       put all updated files into place at end
-m, --prune-empty-dirs  prune empty directory chains from file-list
  --numeric-ids         don't map uid/gid values by user/group name
  --timeout=SECONDS     set I/O timeout in seconds
  --contimeout=SECONDS  set daemon connection timeout in seconds
-I, --ignore-times      don't skip files that match size and time
  --size-only           skip files that match in size
  --modify-window=NUM   compare mod-times with reduced accuracy
-T, --temp-dir=DIR      create temporary files in directory DIR
-y, --fuzzy             find similar file for basis if no dest file
  --compare-dest=DIR    also compare received files relative to DIR
  --copy-dest=DIR       ... and include copies of unchanged files
  --link-dest=DIR       hardlink to files in DIR when unchanged
-z, --compress          compress file data during the transfer
  --compress-level=NUM  explicitly set compression level
  --skip-compress=LIST  skip compressing files with suffix in LIST
-C, --cvs-exclude       auto-ignore files in the same way CVS does
-f, --filter=RULE       add a file-filtering RULE
-F                      same as --filter='dir-merge /rsync-filter'
                      repeated: --filter='- .rsync-filter'
  --exclude=PATTERN     exclude files matching PATTERN
  --exclude-from=FILE   read exclude patterns from FILE
  --include=PATTERN     don't exclude files matching PATTERN
  --include-from=FILE   read include patterns from FILE
  --files-from=FILE     read list of source-file names from FILE
-0, --from0             all *from/filter files are delimited by 0s
-s, --protect-args      no space-splitting; wildcard chars only

```

```

--address=ADDRESS    bind address for outgoing socket to daemon
--port=PORT          specify double-colon alternate port number
--sockopts=OPTIONS   specify custom TCP options
--blocking-io        use blocking I/O for the remote shell
--stats              give some file-transfer stats
-8, --8-bit-output    leave high-bit chars unescaped in output
-h, --human-readable  output numbers in a human-readable format
--progress           show progress during transfer
-P                  same as --partial --progress
-i, --itemize-changes output a change-summary for all updates
--out-format=FORMAT   output updates using the specified FORMAT
--log-file=FILE       log what we're doing to the specified FILE
--log-file-format=FMT log updates using the specified FMT
--password-file=FILE  read daemon-access password from FILE
--list-only           list the files instead of copying them
--bwlimit=KBPS        limit I/O bandwidth; KBytes per second
--write-batch=FILE    write a batched update to FILE
--only-write-batch=FILE like --write-batch but w/o updating dest
--read-batch=FILE     read a batched update from FILE
--protocol=NUM        force an older protocol version to be used
--iconv=CONVERT_SPEC request charset conversion of filenames
--checksum-seed=NUM   set block/file checksum seed (advanced)
-4, --ipv4            prefer IPv4
-6, --ipv6            prefer IPv6
--version            print version number
(-h) --help          show this help (see below for -h comment)

```

Rsync can also be run as a daemon, in which case the following options are accepted:

```

--daemon            run as an rsync daemon
--address=ADDRESS    bind to the specified address
--bwlimit=KBPS       limit I/O bandwidth; KBytes per second
--config=FILE        specify alternate rsyncd.conf file
--no-detach          do not detach from the parent
--port=PORT          listen on alternate port number
--log-file=FILE      override the "log file" setting
--log-file-format=FMT override the "log format" setting
--sockopts=OPTIONS   specify custom TCP options
-v, --verbose        increase verbosity
-4, --ipv4            prefer IPv4
-6, --ipv6            prefer IPv6
-h, --help           show this help (if used after --daemon)

```

OPTIONS

rsync uses the GNU long options package. Many of the command line options have two variants, one short and one long. These are shown below, separated by commas. Some options only have a long variant. The '=' for options that take a parameter is optional; whitespace can be used instead.

--help Print a short help page describing the options available in rsync and exit. For backward-compatibility with older versions of rsync, the help will also be output if you use the **-h** option without any other args.

--version
print the rsync version number and exit.

-v, --verbose

This option increases the amount of information you are given during the transfer. By default, rsync works silently. A single **-v** will give you information about what files are being transferred and a brief summary at the end. Two **-v** flags will give you information on what files are being skipped and slightly more information at the end. More than two **-v** flags should only be used if you are debugging rsync.

Note that the names of the transferred files that are output are done using a default **--out-format** of “%n%L”, which tells you just the name of the file and, if the item is a link, where it points. At the single **-v** level of verbosity, this does not mention when a file gets its attributes changed. If you ask for an itemized list of changed attributes (either **--itemize-changes** or adding “%i” to the **--out-format** setting), the output (on the client) increases to mention all items that are changed in any way. See the **--out-format** option for more details.

-q, --quiet

This option decreases the amount of information you are given during the transfer, notably suppressing information messages from the remote server. This flag is useful when invoking rsync from cron.

--no-motd

This option affects the information that is output by the client at the start of a daemon transfer. This suppresses the message-of-the-day (MOTD) text, but it also affects the list of modules that the daemon sends in response to the “rsync host:” request (due to a limitation in the rsync protocol), so omit this option if you want to request the list of modules from the daemon.

-I, --ignore-times

Normally rsync will skip any files that are already the same size and have the same modification timestamp. This option turns off this “quick check” behavior, causing all files to be updated.

--size-only

This modifies rsync’s “quick check” algorithm for finding files that need to be transferred, changing it from the default of transferring files with either a changed size or a changed last-modified time to just looking for files that have changed in size. This is useful when starting to use rsync after using another mirroring system which may not preserve timestamps exactly.

--modify-window

When comparing two timestamps, rsync treats the timestamps as being equal if they differ by no more than the modify-window value. This is normally 0 (for an exact match), but you may find it useful to set this to a larger value in some situations. In particular, when transferring to or from an MS Windows FAT filesystem (which represents times with a 2-second resolution), **--modify-window=1** is useful (allowing times to differ by up to 1 second).

-c, --checksum

This changes the way rsync checks if the files have been changed and are in need of a transfer. Without this option, rsync uses a “quick check” that (by default) checks if each file’s size and time of last modification match between the sender and receiver. This option changes this to compare a 128-bit MD4 checksum for each file that has a matching size. Generating the checksums means that both sides will expend a lot of disk I/O reading all the data in the files in the transfer (and this is prior to any reading that will be done to transfer changed files), so this can slow things down significantly.

The sending side generates its checksums while it is doing the file-system scan that builds the list of the available files. The receiver generates its checksums when it is scanning for changed files, and will checksum any file that has the same size as the corresponding sender’s file: files with either a changed size or a changed checksum are selected for transfer.

Note that rsync always verifies that each *transferred* file was correctly reconstructed on the receiving side by checking a whole-file checksum that is generated as the file is transferred, but that automatic after-the-transfer verification has nothing to do with this option’s before-the-transfer “Does this file need to be updated?” check.

-a, --archive

This is equivalent to **-rlptgoD**. It is a quick way of saying you want recursion and want to preserve almost everything (with **-H** being a notable omission). The only exception to the above equivalence is when **--files-from** is specified, in which case **-r** is not implied.

Note that **-a** **does not preserve hardlinks**, because finding multiply-linked files is expensive. You must separately specify **-H**.

--no-OPTION

You may turn off one or more implied options by prefixing the option name with “no-”. Not all options may be prefixed with a “no-”: only options that are implied by other options (e.g. **--no-D**, **--no-perms**) or have different defaults in various circumstances (e.g. **--no-whole-file**, **--no-blocking-io**, **--no-dirs**). You may specify either the short or the long option name after the “no-” prefix (e.g. **--no-R** is the same as **--no-relative**).

For example: if you want to use **-a** (**--archive**) but don’t want **-o** (**--owner**), instead of converting **-a** into **-rlptgD**, you could specify **-a --no-o** (or **-a --no-owner**).

The order of the options is important: if you specify **--no-r -a**, the **-r** option would end up being turned on, the opposite of **-a --no-r**. Note also that the side-effects of the **--files-from** option are NOT positional, as it affects the default state of several options and slightly changes the meaning of **-a** (see the **--files-from** option for more details).

-r, --recursive

This tells rsync to copy directories recursively. See also **--dirs (-d)**.

Beginning with rsync 3.0.0, the recursive algorithm used is now an incremental scan that uses much less memory than before and begins the transfer after the scanning of the first few directories have been completed. This incremental scan only affects our recursion algorithm, and does not change a non-recursive transfer. It is also only possible when both ends of the transfer are at least version 3.0.0.

Some options require rsync to know the full file list, so these options disable the incremental recursion mode. These include: **--delete-before**, **--delete-after**, **--prune-empty-dirs**, and **--delay-updates**. Because of this, the default delete mode when you specify **--delete** is now **--delete-during** when both ends of the connection are at least 3.0.0 (use **--del** or **--delete-during** to request this improved deletion mode explicitly). See also the **--delete-delay** option that is a better choice than using **--delete-after**.

Incremental recursion can be disabled using the **--no-inc-recursive** option or its shorter **--no-i-r** alias.

-R, --relative

Use relative paths. This means that the full path names specified on the command line are sent to the server rather than just the last parts of the filenames. This is particularly useful when you want to send several different directories at the same time. For example, if you used this command:

```
rsync -av /foo/bar/baz.c remote:/tmp/
```

... this would create a file named `baz.c` in `/tmp/` on the remote machine. If instead you used

```
rsync -avR /foo/bar/baz.c remote:/tmp/
```

then a file named `/tmp/foo/bar/baz.c` would be created on the remote machine, preserving its full path. These extra path elements are called “implied directories” (i.e. the “foo” and the “foo/bar” directories in the above example).

Beginning with rsync 3.0.0, rsync always sends these implied directories as real directories in the file list, even if a path element is really a symlink on the sending side. This prevents some really unexpected behaviors when copying the full path of a file that you didn’t realize had a symlink in its path. If you want to duplicate a server-side symlink, include both the symlink via its path, and

referent directory via its real path. If you're dealing with an older rsync on the sending side, you may need to use the **--no-implied-dirs** option.

It is also possible to limit the amount of path information that is sent as implied directories for each path you specify. With a modern rsync on the sending side (beginning with 2.6.7), you can insert a dot and a slash into the source path, like this:

```
rsync -avR /foo/./bar/baz.c remote:/tmp/
```

That would create /tmp/bar/baz.c on the remote machine. (Note that the dot must be followed by a slash, so "/foo/." would not be abbreviated.) (2) For older rsync versions, you would need to use a **chdir** to limit the source path. For example, when pushing files:

```
(cd /foo; rsync -avR bar/baz.c remote:/tmp/)
```

(Note that the parens put the two commands into a sub-shell, so that the "cd" command doesn't remain in effect for future commands.) If you're pulling files from an older rsync, use this idiom (but only for a non-daemon transfer):

```
rsync -avR --rsync-path="cd /foo; rsync" \
remote:bar/baz.c /tmp/
```

--no-implied-dirs

This option affects the default behavior of the **--relative** option. When it is specified, the attributes of the implied directories from the source names are not included in the transfer. This means that the corresponding path elements on the destination system are left unchanged if they exist, and any missing implied directories are created with default attributes. This even allows these implied path elements to have big differences, such as being a symlink to a directory on the receiving side.

For instance, if a command-line arg or a files-from entry told rsync to transfer the file "path/foo/file", the directories "path" and "path/foo" are implied when **--relative** is used. If "path/foo" is a symlink to "bar" on the destination system, the receiving rsync would ordinarily delete "path/foo", recreate it as a directory, and receive the file into the new directory. With **--no-implied-dirs**, the receiving rsync updates "path/foo/file" using the existing path elements, which means that the file ends up being created in "path/bar". Another way to accomplish this link preservation is to use the **--keep-dirlinks** option (which will also affect symlinks to directories in the rest of the transfer).

When pulling files from an rsync older than 3.0.0, you may need to use this option if the sending side has a symlink in the path you request and you wish the implied directories to be transferred as normal directories.

-b, --backup

With this option, preexisting destination files are renamed as each file is transferred or deleted. You can control where the backup file goes and what (if any) suffix gets appended using the **--backup-dir** and **--suffix** options.

Note that if you don't specify **--backup-dir**, (1) the **--omit-dir-times** option will be implied, and (2) if **--delete** is also in effect (without **--delete-excluded**), rsync will add a "protect" filter-rule for the backup suffix to the end of all your existing excludes (e.g. **-f "Pp *"**). This will prevent previously backed-up files from being deleted. Note that if you are supplying your own filter rules, you may need to manually insert your own exclude/protect rule somewhere higher up in the list so that it has a high enough priority to be effective (e.g., if your rules specify a trailing inclusion/exclusion of *****, the auto-added rule would never be reached).

--backup-dir=DIR

In combination with the **--backup** option, this tells rsync to store all backups in the specified directory on the receiving side. This can be used for incremental backups. You can additionally

specify a backup suffix using the **--suffix** option (otherwise the files backed up in the specified directory will keep their original filenames).

--suffix=SUFFIX

This option allows you to override the default backup suffix used with the **--backup (-b)** option. The default suffix is a ~ if no **--backup-dir** was specified, otherwise it is an empty string.

-u, --update

This forces rsync to skip any files which exist on the destination and have a modified time that is newer than the source file. (If an existing destination file has a modification time equal to the source file's, it will be updated if the sizes are different.)

Note that this does not affect the copying of symlinks or other special files. Also, a difference of file format between the sender and receiver is always considered to be important enough for an update, no matter what date is on the objects. In other words, if the source has a directory where the destination has a file, the transfer would occur regardless of the timestamps.

--inplace

This causes rsync not to create a new copy of the file and then move it into place. Instead rsync will overwrite the existing file, meaning that the rsync algorithm can't accomplish the full amount of network reduction it might be able to otherwise (since it does not yet try to sort data matches). One exception to this is if you combine the option with **--backup**, since rsync is smart enough to use the backup file as the basis file for the transfer.

This option is useful for transfer of large files with block-based changes or appended data, and also on systems that are disk bound, not network bound.

The option implies **--partial** (since an interrupted transfer does not delete the file), but conflicts with **--partial-dir** and **--delay-updates**. Prior to rsync 2.6.4 **--inplace** was also incompatible with **--compare-dest** and **--link-dest**.

WARNING: The file's data will be in an inconsistent state during the transfer (and possibly afterward if the transfer gets interrupted), so you should not use this option to update files that are in use. Also note that rsync will be unable to update a file in-place that is not writable by the receiving user.

--append

This causes rsync to update a file by appending data onto the end of the file, which presumes that the data that already exists on the receiving side is identical with the start of the file on the sending side. If a file needs to be transferred and its size on the receiver is the same or longer than the size on the sender, the file is skipped. This does not interfere with the updating of a file's non-content attributes (e.g. permissions, ownership, etc.) when the file does not need to be transferred, nor does it affect the updating of any non-regular files. Implies **--inplace**, but does not conflict with **--sparse** (since it is always extending a file's length).

--append-verify

This works just like the **--append** option, but the existing data on the receiving side is included in the full-file checksum verification step, which will cause a file to be resent if the final verification step fails (rsync uses a normal, non-appending **--inplace** transfer for the resend).

Note: prior to rsync 3.0.0, the **--append** option worked like **--append-verify**, so if you are interacting with an older rsync (or the transfer is using a protocol prior to 30), specifying either append option will initiate an **--append-verify** transfer.

-d, --dirs

Tell the sending side to include any directories that are encountered. Unlike **--recursive**, a directory's contents are not copied unless the directory name specified is "." or ends with a trailing slash (e.g. ".", "dir/.", "dir/", etc.). Without this option or the **--recursive** option, rsync will skip all directories it encounters (and output a message to that effect for each one). If you specify both **--dirs** and **--recursive**, **--recursive** takes precedence.

The **--dirs** option is implied by the **--files-from** option or the **--list-only** option (including an implied **--list-only** usage) if **--recursive** wasn't specified (so that directories are seen in the listing). Specify **--no-dirs** (or **--no-d**) if you want to turn this off.

There is also a backward-compatibility helper option, **--old-dirs** (or **--old-d**) that tells rsync to use a hack of "**-r --exclude='/*/*'**" to get an older rsync to list a single directory without recursing.

-l, --links

When symlinks are encountered, recreate the symlink on the destination.

-L, --copy-links

When symlinks are encountered, the item that they point to (the referent) is copied, rather than the symlink. In older versions of rsync, this option also had the side-effect of telling the receiving side to follow symlinks, such as symlinks to directories. In a modern rsync such as this one, you'll need to specify **--keep-dirlinks (-K)** to get this extra behavior. The only exception is when sending files to an rsync that is too old to understand **-K** — in that case, the **-L** option will still have the side-effect of **-K** on that older receiving rsync.

--copy-unsafe-links

This tells rsync to copy the referent of symbolic links that point outside the copied tree. Absolute symlinks are also treated like ordinary files, and so are any symlinks in the source path itself when **--relative** is used. This option has no additional effect if **--copy-links** was also specified.

--safe-links

This tells rsync to ignore any symbolic links which point outside the copied tree. All absolute symlinks are also ignored. Using this option in conjunction with **--relative** may give unexpected results.

-k, --copy-dirlinks

This option causes the sending side to treat a symlink to a directory as though it were a real directory. This is useful if you don't want symlinks to non-directories to be affected, as they would be using **--copy-links**.

Without this option, if the sending side has replaced a directory with a symlink to a directory, the receiving side will delete anything that is in the way of the new symlink, including a directory hierarchy (as long as **--force** or **--delete** is in effect).

See also **--keep-dirlinks** for an analogous option for the receiving side.

-K, --keep-dirlinks

This option causes the receiving side to treat a symlink to a directory as though it were a real directory, but only if it matches a real directory from the sender. Without this option, the receiver's symlink would be deleted and replaced with a real directory.

For example, suppose you transfer a directory "foo" that contains a file "file", but "foo" is a symlink to directory "bar" on the receiver. Without **--keep-dirlinks**, the receiver deletes symlink "foo", recreates it as a directory, and receives the file into the new directory. With **--keep-dirlinks**, the receiver keeps the symlink and "file" ends up in "bar".

One note of caution: if you use **--keep-dirlinks**, you must trust all the symlinks in the copy! If it is possible for an untrusted user to create their own symlink to any directory, the user could then (on a subsequent copy) replace the symlink with a real directory and affect the content of whatever directory the symlink references. For backup copies, you are better off using something like a bind mount instead of a symlink to modify your receiving hierarchy.

See also **--copy-dirlinks** for an analogous option for the sending side.

-H, --hard-links

This tells rsync to look for hard-linked files in the transfer and link together the corresponding files on the receiving side. Without this option, hard-linked files in the transfer are treated as though they were separate files.

When you are updating a non-empty destination, this option only ensures that files that are hard-linked together on the source are hard-linked together on the destination. It does NOT currently endeavor to break already existing hard links on the destination that do not exist between the source files. Note, however, that if one or more extra-linked files have content changes, they will become unlinked when updated (assuming you are not using the **---inplace** option).

Note that rsync can only detect hard links between files that are inside the transfer set. If rsync updates a file that has extra hard-link connections to files outside the transfer, that linkage will be broken. If you are tempted to use the **---inplace** option to avoid this breakage, be very careful that you know how your files are being updated so that you are certain that no unintended changes happen due to lingering hard links (and see the **---inplace** option for more caveats).

If incremental recursion is active (see **---recursive**), rsync may transfer a missing hard-linked file before it finds that another link for that contents exists elsewhere in the hierarchy. This does not affect the accuracy of the transfer, just its efficiency. One way to avoid this is to disable incremental recursion using the **---no-inc-recursive** option.

-p, ---perms

This option causes the receiving rsync to set the destination permissions to be the same as the source permissions. (See also the **---chmod** option for a way to modify what rsync considers to be the source permissions.)

When this option is *off*, permissions are set as follows:

- o Existing files (including updated files) retain their existing permissions, though the **---executability** option might change just the execute permission for the file.
- o New files get their “normal” permission bits set to the source file’s permissions masked with the receiving directory’s default permissions (either the receiving process’s umask, or the permissions specified via the destination directory’s default ACL), and their special permission bits disabled except in the case where a new directory inherits a setgid bit from its parent directory.

Thus, when **---perms** and **---executability** are both disabled, rsync’s behavior is the same as that of other file-copy utilities, such as **cp(1)** and **tar(1)**.

In summary: to give destination files (both old and new) the source permissions, use **---perms**. To give new files the destination-default permissions (while leaving existing files unchanged), make sure that the **---perms** option is off and use **---chmod=ugo=rwX** (which ensures that all non-masked bits get enabled). If you’d care to make this latter behavior easier to type, you could define a popt alias for it, such as putting this line in the file `~/popt` (the following defines the **-Z** option, and includes **---no-g** to use the default group of the destination dir):

```
rsync alias -Z --no-p --no-g --chmod=ugo=rwX
```

You could then use this new option in a command such as this one:

```
rsync -avZ src/ dest/
```

(Caveat: make sure that **-a** does not follow **-Z**, or it will re-enable the two “**---no-***” options mentioned above.)

The preservation of the destination’s setgid bit on newly-created directories when **---perms** is off was added in rsync 2.6.7. Older rsync versions erroneously preserved the three special permission bits for newly-created files when **---perms** was off, while overriding the destination’s setgid bit setting on a newly-created directory. Default ACL observance was added to the ACL patch for rsync 2.6.7, so older (or non-ACL-enabled) rsyncs use the umask even if default ACLs are present. (Keep in mind that it is the version of the receiving rsync that affects these behaviors.)

-E, --executability

This option causes rsync to preserve the executability (or non-executability) of regular files when **--perms** is not enabled. A regular file is considered to be executable if at least one 'x' is turned on in its permissions. When an existing destination file's executability differs from that of the corresponding source file, rsync modifies the destination file's permissions as follows:

- o To make a file non-executable, rsync turns off all its 'x' permissions.
- o To make a file executable, rsync turns on each 'x' permission that has a corresponding 'r' permission enabled.

If **--perms** is enabled, this option is ignored.

-A, --acls

This option causes rsync to update the destination ACLs to be the same as the source ACLs. The option also implies **--perms**.

The source and destination systems must have compatible ACL entries for this option to work properly. See the **--fake-super** option for a way to backup and restore ACLs that are not compatible.

-X, --xattrs

This option causes rsync to update the remote extended attributes to be the same as the local ones.

For systems that support extended-attribute namespaces, a copy being done by a super-user copies all namespaces except system.*. A normal user only copies the user.* namespace. To be able to backup and restore non-user namespaces as a normal user, see the **--fake-super** option.

--chmod

This option tells rsync to apply one or more comma-separated "chmod" strings to the permission of the files in the transfer. The resulting value is treated as though it was the permissions that the sending side supplied for the file, which means that this option can seem to have no effect on existing files if **--perms** is not enabled.

In addition to the normal parsing rules specified in the **chmod(1)** manpage, you can specify an item that should only apply to a directory by prefixing it with a 'D', or specify an item that should only apply to a file by prefixing it with a 'F'. For example:

```
--chmod=Dg+s,ug+w,Fo-w,+X
```

It is also legal to specify multiple **--chmod** options, as each additional option is just appended to the list of changes to make.

See the **--perms** and **--executability** options for how the resulting permission value can be applied to the files in the transfer.

-o, --owner

This option causes rsync to set the owner of the destination file to be the same as the source file, but only if the receiving rsync is being run as the super-user (see also the **--super** and **--fake-super** options). Without this option, the owner of new and/or transferred files are set to the invoking user on the receiving side.

The preservation of ownership will associate matching names by default, but may fall back to using the ID number in some circumstances (see also the **--numeric-ids** option for a full discussion).

-g, --group

This option causes rsync to set the group of the destination file to be the same as the source file. If the receiving program is not running as the super-user (or if **--no-super** was specified), only groups that the invoking user on the receiving side is a member of will be preserved. Without this option, the group is set to the default group of the invoking user on the receiving side.

The preservation of group information will associate matching names by default, but may fall back to using the ID number in some circumstances (see also the **--numeric-ids** option for a full discussion).

--devices

This option causes rsync to transfer character and block device files to the remote system to recreate these devices. This option has no effect if the receiving rsync is not run as the super-user (see also the **--super** and **--fake-super** options).

--specials

This option causes rsync to transfer special files such as named sockets and fifos.

-D The **-D** option is equivalent to **--devices --specials**.

-t, --times

This tells rsync to transfer modification times along with the files and update them on the remote system. Note that if this option is not used, the optimization that excludes files that have not been modified cannot be effective; in other words, a missing **-t** or **-a** will cause the next transfer to behave as if it used **-I**, causing all files to be updated (though the rsync algorithm will make the update fairly efficient if the files haven't actually changed, you're much better off using **-t**).

-O, --omit-dir-times

This tells rsync to omit directories when it is preserving modification times (see **--times**). If NFS is sharing the directories on the receiving side, it is a good idea to use **-O**. This option is inferred if you use **--backup** without **--backup-dir**.

--super

This tells the receiving side to attempt super-user activities even if the receiving rsync wasn't run by the super-user. These activities include: preserving users via the **--owner** option, preserving all groups (not just the current user's groups) via the **--groups** option, and copying devices via the **--devices** option. This is useful for systems that allow such activities without being the super-user, and also for ensuring that you will get errors if the receiving side isn't being running as the super-user. To turn off super-user activities, the super-user can use **--no-super**.

--fake-super

When this option is enabled, rsync simulates super-user activities by saving/restoring the privileged attributes via special extended attributes that are attached to each file (as needed). This includes the file's owner and group (if it is not the default), the file's device info (device & special files are created as empty text files), and any permission bits that we won't allow to be set on the real file (e.g. the real file gets u-s,g-s,o-t for safety) or that would limit the owner's access (since the real super-user can always access/change a file, the files we create can always be accessed/changed by the creating user). This option also handles ACLs (if **--acls** was specified) and non-user extended attributes (if **--xattrs** was specified).

This is a good way to backup data without using a super-user, and to store ACLs from incompatible systems.

The **--fake-super** option only affects the side where the option is used. To affect the remote side of a remote-shell connection, specify an rsync path:

```
rsync -av --rsync-path="rsync --fake-super" /src/ host:/dest/
```

Since there is only one "side" in a local copy, this option affects both the sending and receiving of files. You'll need to specify a copy using "localhost" if you need to avoid this, possibly using the "lsh" shell script (from the support directory) as a substitute for an actual remote shell (see **--rsh**).

This option is overridden by both **--super** and **--no-super**.

See also the "fake super" setting in the daemon's rsyncd.conf file.

-S, --sparse

Try to handle sparse files efficiently so they take up less space on the destination. Conflicts with **--inplace** because it's not possible to overwrite data in a sparse fashion.

NOTE: Don't use this option when the destination is a Solaris "tmpfs" filesystem. It doesn't seem to handle seeks over null regions correctly and ends up corrupting the files.

-n, --dry-run

This makes rsync perform a trial run that doesn't make any changes (and produces mostly the same output as a real run). It is most commonly used in combination with the **-v, --verbose** and/or **-i, --itemize-changes** options to see what an rsync command is going to do before one actually runs it.

The output of **--itemize-changes** is supposed to be exactly the same on a dry run and a subsequent real run (barring intentional trickery and system call failures); if it isn't, that's a bug. Other output is the same to the extent practical, but may differ in some areas. Notably, a dry run does not send the actual data for file transfers, so **--progress** has no effect, the "bytes sent", "bytes received", "literal data", and "matched data" statistics are too small, and the "speedup" value is equivalent to a run where no file transfers are needed.

-W, --whole-file

With this option the delta-transfer algorithm is not used and the whole file is sent as-is instead. The transfer may be faster if this option is used when the bandwidth between the source and destination machines is higher than the bandwidth to disk (especially when the "disk" is actually a networked filesystem). This is the default when both the source and destination are specified as local paths.

-x, --one-file-system

This tells rsync to avoid crossing a filesystem boundary when recursing. This does not limit the user's ability to specify items to copy from multiple filesystems, just rsync's recursion through the hierarchy of each directory that the user specified, and also the analogous recursion on the receiving side during deletion. Also keep in mind that rsync treats a "bind" mount to the same device as being on the same filesystem.

If this option is repeated, rsync omits all mount-point directories from the copy. Otherwise, it includes an empty directory at each mount-point it encounters (using the attributes of the mounted directory because those of the underlying mount-point directory are inaccessible).

If rsync has been told to collapse symlinks (via **--copy-links** or **--copy-unsafe-links**), a symlink to a directory on another device is treated like a mount-point. Symlinks to non-directories are unaffected by this option.

--existing, --ignore-non-existing

This tells rsync to skip creating files (including directories) that do not exist yet on the destination. If this option is combined with the **--ignore-existing** option, no files will be updated (which can be useful if all you want to do is delete extraneous files).

--ignore-existing

This tells rsync to skip updating files that already exist on the destination (this does *not* ignore existing directories, or nothing would get done). See also **--existing**.

This option can be useful for those doing backups using the **--link-dest** option when they need to continue a backup run that got interrupted. Since a **--link-dest** run is copied into a new directory hierarchy (when it is used properly), using **--ignore-existing** will ensure that the already-handled files don't get tweaked (which avoids a change in permissions on the hard-linked files). This does mean that this option is only looking at the existing files in the destination hierarchy itself.

--remove-source-files

This tells rsync to remove from the sending side the files (meaning non-directories) that are a part of the transfer and have been successfully duplicated on the receiving side.

--delete

This tells rsync to delete extraneous files from the receiving side (ones that aren't on the sending side), but only for the directories that are being synchronized. You must have asked rsync to send the whole directory (e.g. "dir" or "dir/") without using a wildcard for the directory's contents (e.g. "dir/*") since the wildcard is expanded by the shell and rsync thus gets a request to transfer individual files, not the files' parent directory. Files that are excluded from transfer are also excluded from being deleted unless you use the **--delete-excluded** option or mark the rules as only matching on the sending side (see the include/exclude modifiers in the FILTER RULES section).

Prior to rsync 2.6.7, this option would have no effect unless **--recursive** was enabled. Beginning with 2.6.7, deletions will also occur when **--dirs (-d)** is enabled, but only for directories whose contents are being copied.

This option can be dangerous if used incorrectly! It is a very good idea to first try a run using the **--dry-run** option (**-n**) to see what files are going to be deleted.

If the sending side detects any I/O errors, then the deletion of any files at the destination will be automatically disabled. This is to prevent temporary filesystem failures (such as NFS errors) on the sending side causing a massive deletion of files on the destination. You can override this with the **--ignore-errors** option.

The **--delete** option may be combined with one of the **--delete-WHEN** options without conflict, as well as **--delete-excluded**. However, if none of the **--delete-WHEN** options are specified, rsync will choose the **--delete-during** algorithm when talking to an rsync 3.0.0 or newer, and the **--delete-before** algorithm when talking to an older rsync. See also **--delete-delay** and **--delete-after**.

--delete-before

Request that the file-deletions on the receiving side be done before the transfer starts. See **--delete** (which is implied) for more details on file-deletion.

Deleting before the transfer is helpful if the filesystem is tight for space and removing extraneous files would help to make the transfer possible. However, it does introduce a delay before the start of the transfer, and this delay might cause the transfer to timeout (if **--timeout** was specified). It also forces rsync to use the old, non-incremental recursion algorithm that requires rsync to scan all the files in the transfer into memory at once (see **--recursive**).

--delete-during, --del

Request that the file-deletions on the receiving side be done incrementally as the transfer happens. This is a faster method than choosing the before- or after-transfer algorithm, but it is only supported beginning with rsync version 2.6.4. See **--delete** (which is implied) for more details on file-deletion.

--delete-delay

Request that the file-deletions on the receiving side be computed during the transfer, and then removed after the transfer completes. If the number of removed files overflows an internal buffer, a temporary file will be created on the receiving side to hold the names (it is removed while open, so you shouldn't see it during the transfer). If the creation of the temporary file fails, rsync will try to fall back to using **--delete-after** (which it cannot do if **--recursive** is doing an incremental scan).

--delete-after

Request that the file-deletions on the receiving side be done after the transfer has completed. This is useful if you are sending new per-directory merge files as a part of the transfer and you want their exclusions to take effect for the delete phase of the current transfer. It also forces rsync to use the old, non-incremental recursion algorithm that requires rsync to scan all the files in the transfer into memory at once (see **--recursive**). See **--delete** (which is implied) for more details on file-deletion.

--delete-excluded

In addition to deleting the files on the receiving side that are not on the sending side, this tells rsync to also delete any files on the receiving side that are excluded (see **--exclude**). See the FILTER RULES section for a way to make individual exclusions behave this way on the receiver, and for a way to protect files from **--delete-excluded**. See **--delete** (which is implied) for more details on file-deletion.

--ignore-errors

Tells **--delete** to go ahead and delete files even when there are I/O errors.

--force

This option tells rsync to delete a non-empty directory when it is to be replaced by a non-directory. This is only relevant if deletions are not active (see **--delete** for details).

Note for older rsync versions: **--force** used to still be required when using **--delete-after**, and it used to be non-functional unless the **--recursive** option was also enabled.

--max-delete=NUM

This tells rsync not to delete more than NUM files or directories. If that limit is exceeded, a warning is output and rsync exits with an error code of 25 (new for 3.0.0).

Also new for version 3.0.0, you may specify **--max-delete=0** to be warned about any extraneous files in the destination without removing any of them. Older clients interpreted this as “unlimited”, so if you don’t know what version the client is, you can use the less obvious **--max-delete=-1** as a backward-compatible way to specify that no deletions be allowed (though older versions didn’t warn when the limit was exceeded).

--max-size=SIZE

This tells rsync to avoid transferring any file that is larger than the specified SIZE. The SIZE value can be suffixed with a string to indicate a size multiplier, and may be a fractional value (e.g. **--max-size=1.5m**).

The suffixes are as follows: “K” (or “KiB”) is a kibibyte (1024), “M” (or “MiB”) is a mebibyte (1024*1024), and “G” (or “GiB”) is a gibibyte (1024*1024*1024). If you want the multiplier to be 1000 instead of 1024, use “KB”, “MB”, or “GB”. (Note: lower-case is also accepted for all values.) Finally, if the suffix ends in either “+1” or “-1”, the value will be offset by one byte in the indicated direction.

Examples: **--max-size=1.5mb-1** is 1499999 bytes, and **--max-size=2g+1** is 2147483649 bytes.

--min-size=SIZE

This tells rsync to avoid transferring any file that is smaller than the specified SIZE, which can help in not transferring small, junk files. See the **--max-size** option for a description of SIZE.

-B, --block-size=BLOCKSIZE

This forces the block size used in the rsync algorithm to a fixed value. It is normally selected based on the size of each file being updated. See the technical report for details.

-e, --rsh=COMMAND

This option allows you to choose an alternative remote shell program to use for communication between the local and remote copies of rsync. Typically, rsync is configured to use ssh by default, but you may prefer to use rsh on a local network.

If this option is used with **[user@]host::module/path**, then the remote shell *COMMAND* will be used to run an rsync daemon on the remote host, and all data will be transmitted through that remote shell connection, rather than through a direct socket connection to a running rsync daemon on the remote host. See the section “USING RSYNC-DAEMON FEATURES VIA A REMOTE-SHELL CONNECTION” above.

Command-line arguments are permitted in *COMMAND* provided that *COMMAND* is presented to rsync as a single argument. You must use spaces (not tabs or other whitespace) to separate the command and args from each other, and you can use single- and/or double-quotes to preserve

spaces in an argument (but not backslashes). Note that doubling a single-quote inside a single-quoted string gives you a single-quote; likewise for double-quotes (though you need to pay attention to which quotes your shell is parsing and which quotes rsync is parsing). Some examples:

```
-e 'ssh -p 2234'
-e 'ssh -o "ProxyCommand nohup ssh firewall nc -w1 %h %p"'
```

(Note that ssh users can alternately customize site-specific connect options in their .ssh/config file.)

You can also choose the remote shell program using the RSYNC_RSH environment variable, which accepts the same range of values as **-e**.

See also the **--blocking-io** option which is affected by this option.

--rsync-path=PROGRAM

Use this to specify what program is to be run on the remote machine to start-up rsync. Often used when rsync is not in the default remote-shell's path (e.g. **--rsync-path=/usr/local/bin/rsync**). Note that PROGRAM is run with the help of a shell, so it can be any program, script, or command sequence you'd care to run, so long as it does not corrupt the standard-in & standard-out that rsync is using to communicate.

One tricky example is to set a different default directory on the remote machine for use with the **--relative** option. For instance:

```
rsync -avR --rsync-path="cd /a/b && rsync" host:c/d /e/
```

-C, --cvs-exclude

This is a useful shorthand for excluding a broad range of files that you often don't want to transfer between systems. It uses a similar algorithm to CVS to determine if a file should be ignored.

The exclude list is initialized to exclude the following items (these initial items are marked as perishable — see the FILTER RULES section):

```
RCS SCCS CVS CVS.adm RCSLOG cvslog.* tags TAGS .make.state
.nse_depinfo *~ #* .* ,* _$* *$ *.old *.bak *.BAK *.orig
*.rej .del-* *.a *.olb *.o *.obj *.so *.exe *.Z *.elc *.ln
core .svn/ .git/ .bzr/
```

then, files listed in a \$HOME/.cvsignore are added to the list and any files listed in the CVSIGNORE environment variable (all cvsignore names are delimited by whitespace).

Finally, any file is ignored if it is in the same directory as a .cvsignore file and matches one of the patterns listed therein. Unlike rsync's filter/exclude files, these patterns are split on whitespace. See the **cvs(1)** manual for more information.

If you're combining **-C** with your own **--filter** rules, you should note that these CVS excludes are appended at the end of your own rules, regardless of where the **-C** was placed on the command-line. This makes them a lower priority than any rules you specified explicitly. If you want to control where these CVS excludes get inserted into your filter rules, you should omit the **-C** as a command-line option and use a combination of **--filter=:C** and **--filter=-C** (either on your command-line or by putting the ":C" and "-C" rules into a filter file with your other rules). The first option turns on the per-directory scanning for the .cvsignore file. The second option does a one-time import of the CVS excludes mentioned above.

-f, --filter=RULE

This option allows you to add rules to selectively exclude certain files from the list of files to be transferred. This is most useful in combination with a recursive transfer.

You may use as many **--filter** options on the command line as you like to build up the list of files to exclude. If the filter contains whitespace, be sure to quote it so that the shell gives the rule to rsync as a single argument. The text below also mentions that you can use an underscore to replace the space that separates a rule from its arg.

See the FILTER RULES section for detailed information on this option.

- F** The **-F** option is a shorthand for adding two **--filter** rules to your command. The first time it is used is a shorthand for this rule:

```
--filter='dir-merge /.rsync-filter'
```

This tells rsync to look for per-directory .rsync-filter files that have been sprinkled through the hierarchy and use their rules to filter the files in the transfer. If **-F** is repeated, it is a shorthand for this rule:

```
--filter='exclude .rsync-filter'
```

This filters out the .rsync-filter files themselves from the transfer.

See the FILTER RULES section for detailed information on how these options work.

--exclude=PATTERN

This option is a simplified form of the **--filter** option that defaults to an exclude rule and does not allow the full rule-parsing syntax of normal filter rules.

See the FILTER RULES section for detailed information on this option.

--exclude-from=FILE

This option is related to the **--exclude** option, but it specifies a FILE that contains exclude patterns (one per line). Blank lines in the file and lines starting with ';' or '#' are ignored. If FILE is -, the list will be read from standard input.

--include=PATTERN

This option is a simplified form of the **--filter** option that defaults to an include rule and does not allow the full rule-parsing syntax of normal filter rules.

See the FILTER RULES section for detailed information on this option.

--include-from=FILE

This option is related to the **--include** option, but it specifies a FILE that contains include patterns (one per line). Blank lines in the file and lines starting with ';' or '#' are ignored. If FILE is -, the list will be read from standard input.

--files-from=FILE

Using this option allows you to specify the exact list of files to transfer (as read from the specified FILE or - for standard input). It also tweaks the default behavior of rsync to make transferring just the specified files and directories easier:

- o The **--relative** (**-R**) option is implied, which preserves the path information that is specified for each item in the file (use **--no-relative** or **--no-R** if you want to turn that off).
- o The **--dirs** (**-d**) option is implied, which will create directories specified in the list on the destination rather than noisily skipping them (use **--no-dirs** or **--no-d** if you want to turn that off).
- o The **--archive** (**-a**) option's behavior does not imply **--recursive** (**-r**), so specify it explicitly, if you want it.
- o These side-effects change the default state of rsync, so the position of the **--files-from** option on the command-line has no bearing on how other options are parsed (e.g. **-a** works the same before or after **--files-from**, as does **--no-R** and all other options).

The filenames that are read from the FILE are all relative to the source dir — any leading slashes are removed and no “..” references are allowed to go higher than the source dir. For example, take this command:

```
rsync -a --files-from=/tmp/foo /usr remote:/backup
```

If /tmp/foo contains the string “bin” (or even “/bin”), the /usr/bin directory will be created as /backup/bin on the remote host. If it contains “bin/” (note the trailing slash), the immediate contents of the directory would also be sent (without needing to be explicitly mentioned in the file — this began in version 2.6.4). In both cases, if the **-r** option was enabled, that dir’s entire hierarchy would also be transferred (keep in mind that **-r** needs to be specified explicitly with **--files-from**, since it is not implied by **-a**). Also note that the effect of the (enabled by default) **--relative** option is to duplicate only the path info that is read from the file — it does not force the duplication of the source-spec path (/usr in this case).

In addition, the **--files-from** file can be read from the remote host instead of the local host if you specify a “host:” in front of the file (the host must match one end of the transfer). As a short-cut, you can specify just a prefix of “:” to mean “use the remote end of the transfer”. For example:

```
rsync -a --files-from=:/path/file-list src:/ /tmp/copy
```

This would copy all the files specified in the /path/file-list file that was located on the remote “src” host.

-0, --from0

This tells rsync that the rules/filenames it reads from a file are terminated by a null ('\0') character, not a NL, CR, or CR+LF. This affects **--exclude-from**, **--include-from**, **--files-from**, and any merged files specified in a **--filter** rule. It does not affect **--cvs-exclude** (since all names read from a .cvsignore file are split on whitespace).

If the **--iconv** and **--protect-args** options are specified and the **--files-from** filenames are being sent from one host to another, the filenames will be translated from the sending host’s charset to the receiving host’s charset.

-s, --protect-args

This option sends all filenames and some options to the remote rsync without allowing the remote shell to interpret them. This means that spaces are not split in names, and any non-wildcard special characters are not translated (such as ~, \$, :, &, etc.). Wildcards are expanded on the remote host by rsync (instead of the shell doing it).

If you use this option with **--iconv**, the args will also be translated from the local to the remote character-set. The translation happens before wild-cards are expanded. See also the **--files-from** option.

-T, --temp-dir=DIR

This option instructs rsync to use DIR as a scratch directory when creating temporary copies of the files transferred on the receiving side. The default behavior is to create each temporary file in the same directory as the associated destination file.

This option is most often used when the receiving disk partition does not have enough free space to hold a copy of the largest file in the transfer. In this case (i.e. when the scratch directory is on a different disk partition), rsync will not be able to rename each received temporary file over the top of the associated destination file, but instead must copy it into place. Rsync does this by copying the file over the top of the destination file, which means that the destination file will contain truncated data during this copy. If this were not done this way (even if the destination file were first removed, the data locally copied to a temporary file in the destination directory, and then renamed into place) it would be possible for the old file to continue taking up disk space (if someone had it open), and thus there might not be enough room to fit the new version on the disk at the same time.

If you are using this option for reasons other than a shortage of disk space, you may wish to combine it with the **--delay-updates** option, which will ensure that all copied files get put into subdirectories in the destination hierarchy, awaiting the end of the transfer. If you don't have enough room to duplicate all the arriving files on the destination partition, another way to tell rsync that you aren't overly concerned about disk space is to use the **--partial-dir** option with a relative path; because this tells rsync that it is OK to stash off a copy of a single file in a subdir in the destination hierarchy, rsync will use the partial-dir as a staging area to bring over the copied file, and then rename it into place from there. (Specifying a **--partial-dir** with an absolute path does not have this side-effect.)

-y, --fuzzy

This option tells rsync that it should look for a basis file for any destination file that is missing. The current algorithm looks in the same directory as the destination file for either a file that has an identical size and modified-time, or a similarly-named file. If found, rsync uses the fuzzy basis file to try to speed up the transfer.

Note that the use of the **--delete** option might get rid of any potential fuzzy-match files, so either use **--delete-after** or specify some filename exclusions if you need to prevent this.

--compare=dest=DIR

This option instructs rsync to use *DIR* on the destination machine as an additional hierarchy to compare destination files against doing transfers (if the files are missing in the destination directory). If a file is found in *DIR* that is identical to the sender's file, the file will NOT be transferred to the destination directory. This is useful for creating a sparse backup of just files that have changed from an earlier backup.

Beginning in version 2.6.4, multiple **--compare=dest** directories may be provided, which will cause rsync to search the list in the order specified for an exact match. If a match is found that differs only in attributes, a local copy is made and the attributes updated. If a match is not found, a basis file from one of the *DIRs* will be selected to try to speed up the transfer.

If *DIR* is a relative path, it is relative to the destination directory. See also **--copy=dest** and **--link=dest**.

--copy=dest=DIR

This option behaves like **--compare=dest**, but rsync will also copy unchanged files found in *DIR* to the destination directory using a local copy. This is useful for doing transfers to a new destination while leaving existing files intact, and then doing a flash-cutover when all files have been successfully transferred.

Multiple **--copy=dest** directories may be provided, which will cause rsync to search the list in the order specified for an unchanged file. If a match is not found, a basis file from one of the *DIRs* will be selected to try to speed up the transfer.

If *DIR* is a relative path, it is relative to the destination directory. See also **--compare=dest** and **--link=dest**.

--link=dest=DIR

This option behaves like **--copy=dest**, but unchanged files are hard linked from *DIR* to the destination directory. The files must be identical in all preserved attributes (e.g. permissions, possibly ownership) in order for the files to be linked together. An example:

```
rsync -av --link=dest=$PWD/prior_dir host:src_dir/ new_dir/
```

Beginning in version 2.6.4, multiple **--link=dest** directories may be provided, which will cause rsync to search the list in the order specified for an exact match. If a match is found that differs only in attributes, a local copy is made and the attributes updated. If a match is not found, a basis file from one of the *DIRs* will be selected to try to speed up the transfer.

This option works best when copying into an empty destination hierarchy, as rsync treats existing files as definitive (so it never looks in the link=dest dirs when a destination file already exists), and

as malleable (so it might change the attributes of a destination file, which affects all the hard-linked versions).

Note that if you combine this option with **--ignore-times**, rsync will not link any files together because it only links identical files together as a substitute for transferring the file, never as an additional check after the file is updated.

If *DIR* is a relative path, it is relative to the destination directory. See also **--compare-dest** and **--copy-dest**.

Note that rsync versions prior to 2.6.1 had a bug that could prevent **--link-dest** from working properly for a non-super-user when **-o** was specified (or implied by **-a**). You can work-around this bug by avoiding the **-o** option when sending to an old rsync.

-z, --compress

With this option, rsync compresses the file data as it is sent to the destination machine, which reduces the amount of data being transmitted — something that is useful over a slow connection.

Note that this option typically achieves better compression ratios than can be achieved by using a compressing remote shell or a compressing transport because it takes advantage of the implicit information in the matching data blocks that are not explicitly sent over the connection.

See the **--skip-compress** option for the default list of file suffixes that will not be compressed.

--compress-level=NUM

Explicitly set the compression level to use (see **--compress**) instead of letting it default. If NUM is non-zero, the **--compress** option is implied.

--skip-compress=LIST

Override the list of file suffixes that will not be compressed. The **LIST** should be one or more file suffixes (without the dot) separated by slashes (/).

You may specify an empty string to indicate that no file should be skipped.

Simple character-class matching is supported: each must consist of a list of letters inside the square brackets (e.g. no special classes, such as “[:alpha:]”, are supported).

The characters asterisk (*) and question-mark (?) have no special meaning.

Here’s an example that specifies 6 suffixes to skip (since 1 of the 5 rules matches 2 suffixes):

```
--skip-compress=gz/jpg/mp[34]/7z/bz2
```

The default list of suffixes that will not be compressed is this (several of these are newly added for 3.0.0):

```
gz/zip/z/rpm/deb/iso/bz2/t[gb]z/7z/mp[34]/mov/avi/ogg/jpg/jpeg
```

This list will be replaced by your **--skip-compress** list in all but one situation: a copy from a daemon rsync will add your skipped suffixes to its list of non-compressing files (and its list may be configured to a different default).

--numeric-ids

With this option rsync will transfer numeric group and user IDs rather than using user and group names and mapping them at both ends.

By default rsync will use the username and groupname to determine what ownership to give files. The special uid 0 and the special group 0 are never mapped via user/group names even if the **--numeric-ids** option is not specified.

If a user or group has no name on the source system or it has no match on the destination system, then the numeric ID from the source system is used instead. See also the comments on the “use chroot” setting in the rsyncd.conf manpage for information on how the chroot setting affects rsync’s ability to look up the names of the users and groups and what you can do about it.

--timeout=TIMEOUT

This option allows you to set a maximum I/O timeout in seconds. If no data is transferred for the specified time then rsync will exit. The default is 0, which means no timeout.

--contimeout

This option allows you to set the amount of time that rsync will wait for its connection to an rsync daemon to succeed. If the timeout is reached, rsync exits with an error.

--address

By default rsync will bind to the wildcard address when connecting to an rsync daemon. The **--address** option allows you to specify a specific IP address (or hostname) to bind to. See also this option in the **--daemon** mode section.

--port=PORT

This specifies an alternate TCP port number to use rather than the default of 873. This is only needed if you are using the double-colon (::) syntax to connect with an rsync daemon (since the URL syntax has a way to specify the port as a part of the URL). See also this option in the **--daemon** mode section.

--sockopts

This option can provide endless fun for people who like to tune their systems to the utmost degree. You can set all sorts of socket options which may make transfers faster (or slower!). Read the man page for the `setsockopt()` system call for details on some of the options you may be able to set. By default no special socket options are set. This only affects direct socket connections to a remote rsync daemon. This option also exists in the **--daemon** mode section.

--blocking-io

This tells rsync to use blocking I/O when launching a remote shell transport. If the remote shell is either rsh or remsh, rsync defaults to using blocking I/O, otherwise it defaults to using non-blocking I/O. (Note that ssh prefers non-blocking I/O.)

-i, --itemize-changes

Requests a simple itemized list of the changes that are being made to each file, including attribute changes. This is exactly the same as specifying **--out-format='%i %n%L'**. If you repeat the option, unchanged files will also be output, but only if the receiving rsync is at least version 2.6.7 (you can use **-vv** with older versions of rsync, but that also turns on the output of other verbose messages).

The “%i” escape has a cryptic output that is 11 letters long. The general format is like the string **YXcstpoguax**, where **Y** is replaced by the type of update being done, **X** is replaced by the file-type, and the other letters represent attributes that may be output if they are being modified.

The update types that replace the **Y** are as follows:

- o A < means that a file is being transferred to the remote host (sent).
- o A > means that a file is being transferred to the local host (received).
- o A **c** means that a local change/creation is occurring for the item (such as the creation of a directory or the changing of a symlink, etc.).
- o A **h** means that the item is a hard link to another item (requires **--hard-links**).
- o A **.** means that the item is not being updated (though it might have attributes that are being modified).
- o A ***** means that the rest of the itemized-output area contains a message (e.g. “deleting”).

The file-types that replace the **X** are: **f** for a file, a **d** for a directory, an **L** for a symlink, a **D** for a device, and a **S** for a special file (e.g. named sockets and fifos).

The other letters in the string above are the actual letters that will be output if the associated attribute for the item is being updated or a “.” for no change. Three exceptions to this are: (1) a

newly created item replaces each letter with a “+”, (2) an identical item replaces the dots with spaces, and (3) an unknown attribute replaces each letter with a “?” (this can happen when talking to an older rsync).

The attribute that is associated with each letter is as follows:

- o A **c** means either that a regular file has a different checksum (requires **--checksum**) or that a symlink, device, or special file has a changed value. Note that if you are sending files to an rsync prior to 3.0.1, this change flag will be present only for checksum-differing regular files.
- o A **s** means the size of a regular file is different and will be updated by the file transfer.
- o A **t** means the modification time is different and is being updated to the sender’s value (requires **--times**). An alternate value of **T** means that the modification time will be set to the transfer time, which happens when a file/symlink/device is updated without **--times** and when a symlink is changed and the receiver can’t set its time. (Note: when using an rsync 3.0.0 client, you might see the **s** flag combined with **t** instead of the proper **T** flag for this time-setting failure.)
- o A **p** means the permissions are different and are being updated to the sender’s value (requires **--perms**).
- o An **o** means the owner is different and is being updated to the sender’s value (requires **--owner** and super-user privileges).
- o A **g** means the group is different and is being updated to the sender’s value (requires **--group** and the authority to set the group).
- o The **u** slot is reserved for future use.
- o The **a** means that the ACL information changed.
- o The **x** means that the extended attribute information changed.

One other output is possible: when deleting files, the “%i” will output the string “*deleting” for each item that is being removed (assuming that you are talking to a recent enough rsync that it logs deletions instead of outputting them as a verbose message).

--out-format=FORMAT

This allows you to specify exactly what the rsync client outputs to the user on a per-update basis. The format is a text string containing embedded single-character escape sequences prefixed with a percent (%) character. For a list of the possible escape characters, see the “log format” setting in the `rsyncd.conf` manpage.

Specifying this option will mention each file, dir, etc. that gets updated in a significant way (a transferred file, a recreated symlink/device, or a touched directory). In addition, if the `itemize-changes` escape (%i) is included in the string, the logging of names increases to mention any item that is changed in any way (as long as the receiving side is at least 2.6.4). See the **--itemize-changes** option for a description of the output of “%i”.

The **--verbose** option implies a format of “%n%L”, but you can use **--out-format** without **--verbose** if you like, or you can override the format of its per-file output using this option.

Rsync will output the out-format string prior to a file’s transfer unless one of the transfer-statistic escapes is requested, in which case the logging is done at the end of the file’s transfer. When this late logging is in effect and **--progress** is also specified, rsync will also output the name of the file being transferred prior to its progress information (followed, of course, by the out-format output).

--log-file=FILE

This option causes rsync to log what it is doing to a file. This is similar to the logging that a daemon does, but can be requested for the client side and/or the server side of a non-daemon transfer.

If specified as a client option, transfer logging will be enabled with a default format of “%i%n%L”. See the **--log-file-format** option if you wish to override this.

Here’s a example command that requests the remote side to log what is happening:

```
rsync -av --rsync-path="rsync --log-file=/tmp/rlog" src/ dest/
```

This is very useful if you need to debug why a connection is closing unexpectedly.

--log-file-format=FORMAT

This allows you to specify exactly what per-update logging is put into the file specified by the **--log-file** option (which must also be specified for this option to have any effect). If you specify an empty string, updated files will not be mentioned in the log file. For a list of the possible escape characters, see the “log format” setting in the rsyncd.conf manpage.

--stats This tells rsync to print a verbose set of statistics on the file transfer, allowing you to tell how effective the rsync algorithm is for your data.

The current statistics are as follows:

- o **Number of files** is the count of all “files” (in the generic sense), which includes directories, symlinks, etc.
- o **Number of files transferred** is the count of normal files that were updated via the rsync algorithm, which does not include created dirs, symlinks, etc.
- o **Total file size** is the total sum of all file sizes in the transfer. This does not count any size for directories or special files, but does include the size of symlinks.
- o **Total transferred file size** is the total sum of all files sizes for just the transferred files.
- o **Literal data** is how much unmatched file-update data we had to send to the receiver for it to recreate the updated files.
- o **Matched data** is how much data the receiver got locally when recreating the updated files.
- o **File list size** is how big the file-list data was when the sender sent it to the receiver. This is smaller than the in-memory size for the file list due to some compressing of duplicated data when rsync sends the list.
- o **File list generation time** is the number of seconds that the sender spent creating the file list. This requires a modern rsync on the sending side for this to be present.
- o **File list transfer time** is the number of seconds that the sender spent sending the file list to the receiver.
- o **Total bytes sent** is the count of all the bytes that rsync sent from the client side to the server side.
- o **Total bytes received** is the count of all non-message bytes that rsync received by the client side from the server side. “Non-message” bytes means that we don’t count the bytes for a verbose message that the server sent to us, which makes the stats more consistent.

-8, --8-bit-output

This tells rsync to leave all high-bit characters unescaped in the output instead of trying to test them to see if they’re valid in the current locale and escaping the invalid ones. All control characters (but never tabs) are always escaped, regardless of this option’s setting.

The escape idiom that started in 2.6.7 is to output a literal backslash (\) and a hash (#), followed by exactly 3 octal digits. For example, a newline would output as “\#012”. A literal backslash that is in a filename is not escaped unless it is followed by a hash and 3 digits (0–9).

-h, --human-readable

Output numbers in a more human-readable format. This makes big numbers output using larger units, with a K, M, or G suffix. If this option was specified once, these units are K (1000), M (1000*1000), and G (1000*1000*1000); if the option is repeated, the units are powers of 1024 instead of 1000.

--partial

By default, rsync will delete any partially transferred file if the transfer is interrupted. In some circumstances it is more desirable to keep partially transferred files. Using the **--partial** option tells rsync to keep the partial file which should make a subsequent transfer of the rest of the file much faster.

--partial-dir=DIR

A better way to keep partial files than the **--partial** option is to specify a *DIR* that will be used to hold the partial data (instead of writing it out to the destination file). On the next transfer, rsync will use a file found in this dir as data to speed up the resumption of the transfer and then delete it after it has served its purpose.

Note that if **--whole-file** is specified (or implied), any partial-dir file that is found for a file that is being updated will simply be removed (since rsync is sending files without using the delta transfer algorithm).

Rsync will create the *DIR* if it is missing (just the last dir — not the whole path). This makes it easy to use a relative path (such as “**--partial-dir=.rsync-partial**”) to have rsync create the partial-directory in the destination file’s directory when needed, and then remove it again when the partial file is deleted.

If the partial-dir value is not an absolute path, rsync will add an exclude rule at the end of all your existing excludes. This will prevent the sending of any partial-dir files that may exist on the sending side, and will also prevent the untimely deletion of partial-dir items on the receiving side. An example: the above **--partial-dir** option would add the equivalent of “**-f ’-p .rsync-partial/**” at the end of any other filter rules.

If you are supplying your own exclude rules, you may need to add your own exclude/hide/protect rule for the partial-dir because (1) the auto-added rule may be ineffective at the end of your other rules, or (2) you may wish to override rsync’s exclude choice. For instance, if you want to make rsync clean-up any left-over partial-dirs that may be lying around, you should specify **--delete-after** and add a “risk” filter rule, e.g. **-f ’R .rsync-partial/**. (Avoid using **--delete-before** or **--delete-during** unless you don’t need rsync to use any of the left-over partial-dir data during the current run.)

IMPORTANT: the **--partial-dir** should not be writable by other users or it is a security risk. E.g. AVOID “/tmp”.

You can also set the partial-dir value the `RSYNC_PARTIAL_DIR` environment variable. Setting this in the environment does not force **--partial** to be enabled, but rather it affects where partial files go when **--partial** is specified. For instance, instead of using **--partial-dir=.rsync-tmp** along with **--progress**, you could set `RSYNC_PARTIAL_DIR=.rsync-tmp` in your environment and then just use the **-P** option to turn on the use of the `.rsync-tmp` dir for partial transfers. The only times that the **--partial** option does not look for this environment value are (1) when **--inplace** was specified (since **--inplace** conflicts with **--partial-dir**), and (2) when **--delay-updates** was specified (see below).

For the purposes of the daemon-config’s “refuse options” setting, **--partial-dir** does *not* imply **--partial**. This is so that a refusal of the **--partial** option can be used to disallow the overwriting of destination files with a partial transfer, while still allowing the safer idiom provided by **--partial-dir**.

--delay-updates

This option puts the temporary file from each updated file into a holding directory until the end of the transfer, at which time all the files are renamed into place in rapid succession. This attempts to make the updating of the files a little more atomic. By default the files are placed into a directory named “`~tmp`” in each file’s destination directory, but if you’ve specified the **--partial-dir** option, that directory will be used instead. See the comments in the **--partial-dir** section for a discussion of how this “`~tmp`” dir will be excluded from the transfer, and what you can do if you want rsync to cleanup old “`~tmp`” dirs that might be lying around. Conflicts with **--inplace** and **--append**.

This option uses more memory on the receiving side (one bit per file transferred) and also requires enough free disk space on the receiving side to hold an additional copy of all the updated files. Note also that you should not use an absolute path to **--partial-dir** unless (1) there is no chance of any of the files in the transfer having the same name (since all the updated files will be put into a single directory if the path is absolute) and (2) there are no mount points in the hierarchy (since the delayed updates will fail if they can’t be renamed into place).

See also the “atomic-rsync” perl script in the “support” subdir for an update algorithm that is even more atomic (it uses **--link-dest** and a parallel hierarchy of files).

-m, --prune-empty-dirs

This option tells the receiving rsync to get rid of empty directories from the file-list, including nested directories that have no non-directory children. This is useful for avoiding the creation of a bunch of useless directories when the sending rsync is recursively scanning a hierarchy of files using include/exclude/filter rules.

Because the file-list is actually being pruned, this option also affects what directories get deleted when a delete is active. However, keep in mind that excluded files and directories can prevent existing items from being deleted (because an exclude hides source files and protects destination files).

You can prevent the pruning of certain empty directories from the file-list by using a global “protect” filter. For instance, this option would ensure that the directory “emptydir” was kept in the file-list:

```
--filter 'protect emptydir/'
```

Here’s an example that copies all .pdf files in a hierarchy, only creating the necessary destination directories to hold the .pdf files, and ensures that any superfluous files and directories in the destination are removed (note the hide filter of non-directories being used instead of an exclude):

```
rsync -avm --del --include='*.pdf' -f 'hide,! */' src/ dest
```

If you didn’t want to remove superfluous destination files, the more time-honored options of “**--include='*/'** **--exclude='*'**” would work fine in place of the hide-filter (if that is more natural to you).

--progress

This option tells rsync to print information showing the progress of the transfer. This gives a bored user something to watch. Implies **--verbose** if it wasn’t already specified.

While rsync is transferring a regular file, it updates a progress line that looks like this:

```
782448 63% 110.64kB/s 0:00:04
```

In this example, the receiver has reconstructed 782448 bytes or 63% of the sender’s file, which is being reconstructed at a rate of 110.64 kilobytes per second, and the transfer will finish in 4 seconds if the current rate is maintained until the end.

These statistics can be misleading if the delta transfer algorithm is in use. For example, if the sender's file consists of the basis file followed by additional data, the reported rate will probably drop dramatically when the receiver gets to the literal data, and the transfer will probably take much longer to finish than the receiver estimated as it was finishing the matched part of the file.

When the file transfer finishes, rsync replaces the progress line with a summary line that looks like this:

```
1238099 100% 146.38kB/s 0:00:08 (xfer#5, to-check=169/396)
```

In this example, the file was 1238099 bytes long in total, the average rate of transfer for the whole file was 146.38 kilobytes per second over the 8 seconds that it took to complete, it was the 5th transfer of a regular file during the current rsync session, and there are 169 more files for the receiver to check (to see if they are up-to-date or not) remaining out of the 396 total files in the file-list.

-P The **-P** option is equivalent to **--partial --progress**. Its purpose is to make it much easier to specify these two options for a long transfer that may be interrupted.

--password=file

This option allows you to provide a password in a file for accessing an rsync daemon. The file must not be world readable. It should contain just the password as a single line.

This option does not supply a password to a remote shell transport such as ssh; to learn how to do that, consult the remote shell's documentation. When accessing an rsync daemon using a remote shell as the transport, this option only comes into effect after the remote shell finishes its authentication (i.e. if you have also specified a password in the daemon's config file).

--list-only

This option will cause the source files to be listed instead of transferred. This option is inferred if there is a single source arg and no destination specified, so its main uses are: (1) to turn a copy command that includes a destination arg into a file-listing command, or (2) to be able to specify more than one source arg (note: be sure to include the destination). Caution: keep in mind that a source arg with a wild-card is expanded by the shell into multiple args, so it is never safe to try to list such an arg without using this option. For example:

```
rsync -av --list-only foo* dest/
```

Compatibility note: when requesting a remote listing of files from an rsync that is version 2.6.3 or older, you may encounter an error if you ask for a non-recursive listing. This is because a file listing implies the **--dirs** option w/o **--recursive**, and older rsyncs don't have that option. To avoid this problem, either specify the **--no-dirs** option (if you don't need to expand a directory's content), or turn on recursion and exclude the content of subdirectories: **-r --exclude='/*/*'**.

--bwlimit=KBPS

This option allows you to specify a maximum transfer rate in kilobytes per second. This option is most effective when using rsync with large files (several megabytes and up). Due to the nature of rsync transfers, blocks of data are sent, then if rsync determines the transfer was too fast, it will wait before sending the next data block. The result is an average transfer rate equaling the specified limit. A value of zero specifies no limit.

--write-batch=FILE

Record a file that can later be applied to another identical destination with **--read-batch**. See the "BATCH MODE" section for details, and also the **--only-write-batch** option.

--only-write-batch=FILE

Works like **--write-batch**, except that no updates are made on the destination system when creating the batch. This lets you transport the changes to the destination system via some other means and then apply the changes via **--read-batch**.

Note that you can feel free to write the batch directly to some portable media: if this media fills to capacity before the end of the transfer, you can just apply that partial transfer to the destination and repeat the whole process to get the rest of the changes (as long as you don't mind a partially updated destination system while the multi-update cycle is happening).

Also note that you only save bandwidth when pushing changes to a remote system because this allows the batched data to be diverted from the sender into the batch file without having to flow over the wire to the receiver (when pulling, the sender is remote, and thus can't write the batch).

--read=batch=FILE

Apply all of the changes stored in FILE, a file previously generated by **--write=batch**. If FILE is -, the batch data will be read from standard input. See the "BATCH MODE" section for details.

--protocol=NUM

Force an older protocol version to be used. This is useful for creating a batch file that is compatible with an older version of rsync. For instance, if rsync 2.6.4 is being used with the **--write=batch** option, but rsync 2.6.3 is what will be used to run the **--read=batch** option, you should use "**--protocol=28**" when creating the batch file to force the older protocol version to be used in the batch file (assuming you can't upgrade the rsync on the reading system).

--iconv=CONVERT_SPEC

Rsync can convert filenames between character sets using this option. Using a CONVERT_SPEC of "." tells rsync to look up the default character-set via the locale setting. Alternately, you can fully specify what conversion to do by giving a local and a remote charset separated by a comma in the order **--iconv=LOCAL,REMOTE**, e.g. **--iconv=utf8,iso88591**. This order ensures that the option will stay the same whether you're pushing or pulling files. Finally, you can specify either **--no-iconv** or a CONVERT_SPEC of "-" to turn off any conversion. The default setting of this option is site-specific, and can also be affected via the RSYNC_ICONV environment variable.

For a list of what charset names your local iconv library supports, you can run "iconv --list".

If you specify the **--protect-args** option (**-s**), rsync will translate the filenames you specify on the command-line that are being sent to the remote host. See also the **--files-from** option.

Note that rsync does not do any conversion of names in filter files (including include/exclude files). It is up to you to ensure that you're specifying matching rules that can match on both sides of the transfer. For instance, you can specify extra include/exclude rules if there are filename differences on the two sides that need to be accounted for.

When you pass an **--iconv** option to an rsync daemon that allows it, the daemon uses the charset specified in its "charset" configuration parameter regardless of the remote charset you actually pass. Thus, you may feel free to specify just the local charset for a daemon transfer (e.g. **--iconv=utf8**).

-4, --ipv4 or -6, --ipv6

Tells rsync to prefer IPv4/IPv6 when creating sockets. This only affects sockets that rsync has direct control over, such as the outgoing socket when directly contacting an rsync daemon. See also these options in the **--daemon** mode section.

If rsync was compiled without support for IPv6, the **--ipv6** option will have no effect. The **--version** output will tell you if this is the case.

--checksum-seed=NUM

Set the MD4 checksum seed to the integer NUM. This 4 byte checksum seed is included in each block and file MD4 checksum calculation. By default the checksum seed is generated by the server and defaults to the current `time()`. This option is used to set a specific checksum seed, which is useful for applications that want repeatable block and file checksums, or in the case where the user wants a more random checksum seed. Note that setting NUM to 0 causes rsync to use the default of `time()` for checksum seed.

DAEMON OPTIONS

The options allowed when starting an rsync daemon are as follows:

--daemon

This tells rsync that it is to run as a daemon. The daemon you start running may be accessed using an rsync client using the **host::module** or **rsync://host/module/** syntax.

If standard input is a socket then rsync will assume that it is being run via inetd, otherwise it will detach from the current terminal and become a background daemon. The daemon will read the config file (rsyncd.conf) on each connect made by a client and respond to requests accordingly. See the **rsyncd.conf(5)** man page for more details.

--address

By default rsync will bind to the wildcard address when run as a daemon with the **--daemon** option. The **--address** option allows you to specify a specific IP address (or hostname) to bind to. This makes virtual hosting possible in conjunction with the **--config** option. See also the “address” global option in the rsyncd.conf manpage.

--bwlimit=KBPS

This option allows you to specify a maximum transfer rate in kilobytes per second for the data the daemon sends. The client can still specify a smaller **--bwlimit** value, but their requested value will be rounded down if they try to exceed it. See the client version of this option (above) for some extra details.

--config=FILE

This specifies an alternate config file than the default. This is only relevant when **--daemon** is specified. The default is /etc/rsyncd.conf unless the daemon is running over a remote shell program and the remote user is not the super-user; in that case the default is rsyncd.conf in the current directory (typically \$HOME).

--no-detach

When running as a daemon, this option instructs rsync to not detach itself and become a background process. This option is required when running as a service on Cygwin, and may also be useful when rsync is supervised by a program such as **daemontools** or AIX’s **System Resource Controller**. **--no-detach** is also recommended when rsync is run under a debugger. This option has no effect if rsync is run from inetd or sshd.

--port=PORT

This specifies an alternate TCP port number for the daemon to listen on rather than the default of 873. See also the “port” global option in the rsyncd.conf manpage.

--log-file=FILE

This option tells the rsync daemon to use the given log-file name instead of using the “log file” setting in the config file.

--log-file-format=FORMAT

This option tells the rsync daemon to use the given FORMAT string instead of using the “log format” setting in the config file. It also enables “transfer logging” unless the string is empty, in which case transfer logging is turned off.

--sockopts

This overrides the **socket options** setting in the rsyncd.conf file and has the same syntax.

-v, --verbose

This option increases the amount of information the daemon logs during its startup phase. After the client connects, the daemon’s verbosity level will be controlled by the options that the client used and the “max verbosity” setting in the module’s config section.

-4, --ipv4 or -6, --ipv6

Tells rsync to prefer IPv4/IPv6 when creating the incoming sockets that the rsync daemon will use to listen for connections. One of these options may be required in older versions of Linux to work around an IPv6 bug in the kernel (if you see an “address already in use” error when nothing else is

using the port, try specifying **--ipv6** or **--ipv4** when starting the daemon).

If rsync was compiled without support for IPv6, the **--ipv6** option will have no effect. The **--version** output will tell you if this is the case.

-h, --help

When specified after **--daemon**, print a short help page describing the options available for starting an rsync daemon.

FILTER RULES

The filter rules allow for flexible selection of which files to transfer (include) and which files to skip (exclude). The rules either directly specify include/exclude patterns or they specify a way to acquire more include/exclude patterns (e.g. to read them from a file).

As the list of files/directories to transfer is built, rsync checks each name to be transferred against the list of include/exclude patterns in turn, and the first matching pattern is acted on: if it is an exclude pattern, then that file is skipped; if it is an include pattern then that filename is not skipped; if no matching pattern is found, then the filename is not skipped.

Rsync builds an ordered list of filter rules as specified on the command-line. Filter rules have the following syntax:

```
RULE [PATTERN_OR_FILENAME]
RULE, MODIFIERS [PATTERN_OR_FILENAME]
```

You have your choice of using either short or long RULE names, as described below. If you use a short-named rule, the ‘,’ separating the RULE from the MODIFIERS is optional. The PATTERN or FILENAME that follows (when present) must come after either a single space or an underscore (_). Here are the available rule prefixes:

- exclude, -** specifies an exclude pattern.
- include, +** specifies an include pattern.
- merge, .** specifies a merge-file to read for more rules.
- dir-merge, :** specifies a per-directory merge-file.
- hide, H** specifies a pattern for hiding files from the transfer.
- show, S** files that match the pattern are not hidden.
- protect, P** specifies a pattern for protecting files from deletion.
- risk, R** files that match the pattern are not protected.
- clear, !** clears the current include/exclude list (takes no arg)

When rules are being read from a file, empty lines are ignored, as are comment lines that start with a “#”.

Note that the **--include/--exclude** command-line options do not allow the full range of rule parsing as described above — they only allow the specification of include/exclude patterns plus a “!” token to clear the list (and the normal comment parsing when rules are read from a file). If a pattern does not begin with “- ” (dash, space) or “+ ” (plus, space), then the rule will be interpreted as if “+ ” (for an include option) or “- ” (for an exclude option) were prefixed to the string. A **--filter** option, on the other hand, must always contain either a short or long rule name at the start of the rule.

Note also that the **--filter**, **--include**, and **--exclude** options take one rule/pattern each. To add multiple ones, you can repeat the options on the command-line, use the merge-file syntax of the **--filter** option, or the **--include-from/--exclude-from** options.

INCLUDE/EXCLUDE PATTERN RULES

You can include and exclude files by specifying patterns using the “+”, “-”, etc. filter rules (as introduced in the FILTER RULES section above). The include/exclude rules each specify a pattern that is matched against the names of the files that are going to be transferred. These patterns can take several forms:

- o if the pattern starts with a / then it is anchored to a particular spot in the hierarchy of files, otherwise it is matched against the end of the pathname. This is similar to a leading ^ in regular expressions. Thus “/foo” would match a name of “foo” at either the “root of the transfer” (for a global rule) or in the merge-file’s directory (for a per-directory rule). An unqualified “foo” would match a name of “foo” anywhere in the tree because the algorithm is applied recursively from the top down; it behaves as if each path component gets a turn at being the end of the filename. Even the unanchored “sub/foo” would match at any point in the hierarchy where a “foo” was found within a directory named “sub”. See the section on ANCHORING INCLUDE/EXCLUDE PATTERNS for a full discussion of how to specify a pattern that matches at the root of the transfer.
- o if the pattern ends with a / then it will only match a directory, not a regular file, symlink, or device.
- o rsync chooses between doing a simple string match and wildcard matching by checking if the pattern contains one of these three wildcard characters: ‘*’, ‘?’, and ‘[’.
- o a ‘*’ matches any non-empty path component (it stops at slashes).
- o use ‘***’ to match anything, including slashes.
- o a ‘?’ matches any character except a slash (/).
- o a ‘[’ introduces a character class, such as [a–z] or [[:alpha:]].
- o in a wildcard pattern, a backslash can be used to escape a wildcard character, but it is matched literally when no wildcards are present.
- o if the pattern contains a / (not counting a trailing /) or a “***”, then it is matched against the full pathname, including any leading directories. If the pattern doesn’t contain a / or a “***”, then it is matched only against the final component of the filename. (Remember that the algorithm is applied recursively so “full filename” can actually be any portion of a path from the starting directory on down.)
- o a trailing “dir_name/***” will match both the directory (as if “dir_name/” had been specified) and everything in the directory (as if “dir_name/***” had been specified). This behavior was added in version 2.6.7.

Note that, when using the **—recursive** (**–r**) option (which is implied by **–a**), every subcomponent of every path is visited from the top down, so include/exclude patterns get applied recursively to each subcomponent’s full name (e.g. to include “/foo/bar/baz” the subcomponents “/foo” and “/foo/bar” must not be excluded). The exclude patterns actually short-circuit the directory traversal stage when rsync finds the files to send. If a pattern excludes a particular parent directory, it can render a deeper include pattern ineffectual because rsync did not descend through that excluded section of the hierarchy. This is particularly important when using a trailing ‘*’ rule. For instance, this won’t work:

```
+ /some/path/this-file-will-not-be-found
+ /file-is-included
- *
```

This fails because the parent directory “some” is excluded by the ‘*’ rule, so rsync never visits any of the files in the “some” or “some/path” directories. One solution is to ask for all directories in the hierarchy to be included by using a single rule: “+ */” (put it somewhere before the “– *” rule), and perhaps use the **—prune-empty-dirs** option. Another solution is to add specific include rules for all the parent dirs that need to be visited. For instance, this set of rules works fine:

```
+ /some/
+ /some/path/
+ /some/path/this-file-is-found
+ /file-also-included
- *
```

Here are some examples of exclude/include matching:

- o “- *.o” would exclude all names matching *.o
- o “- /foo” would exclude a file (or directory) named foo in the transfer-root directory
- o “- foo/” would exclude any directory named foo
- o “- /foo/*/bar” would exclude any file named bar which is at two levels below a directory named foo in the transfer-root directory
- o “- /foo/**/bar” would exclude any file named bar two or more levels below a directory named foo in the transfer-root directory
- o The combination of “+ */”, “+ *.c”, and “- *” would include all directories and C source files but nothing else (see also the **--prune-empty-dirs** option)
- o The combination of “+ foo/”, “+ foo/bar.c”, and “- *” would include only the foo directory and foo/bar.c (the foo directory must be explicitly included or it would be excluded by the “-”)

MERGE-FILE FILTER RULES

You can merge whole files into your filter rules by specifying either a merge (.) or a dir-merge (:) filter rule (as introduced in the FILTER RULES section above).

There are two kinds of merged files — single-instance (‘.’) and per-directory (‘:’). A single-instance merge file is read one time, and its rules are incorporated into the filter list in the place of the “.” rule. For per-directory merge files, rsync will scan every directory that it traverses for the named file, merging its contents when the file exists into the current list of inherited rules. These per-directory rule files must be created on the sending side because it is the sending side that is being scanned for the available files to transfer. These rule files may also need to be transferred to the receiving side if you want them to affect what files don’t get deleted (see PER-DIRECTORY RULES AND DELETE below).

Some examples:

```
merge /etc/rsync/default.rules
. /etc/rsync/default.rules
dir-merge .per-dir-filter
dir-merge,n- .non-inherited-per-dir-excludes
:n- .non-inherited-per-dir-excludes
```

The following modifiers are accepted after a merge or dir-merge rule:

- o A **-** specifies that the file should consist of only exclude patterns, with no other rule-parsing except for in-file comments.
- o A **+** specifies that the file should consist of only include patterns, with no other rule-parsing except for in-file comments.
- o A **C** is a way to specify that the file should be read in a CVS-compatible manner. This turns on ‘n’, ‘w’, and ‘-’, but also allows the list-clearing token (!) to be specified. If no filename is provided, “cvsignore” is assumed.
- o A **e** will exclude the merge-file name from the transfer; e.g. “dir-merge,e.rules” is like “dir-merge.rules” and “- .rules”.
- o An **n** specifies that the rules are not inherited by subdirectories.
- o A **w** specifies that the rules are word-split on whitespace instead of the normal line-splitting. This also turns off comments. Note: the space that separates the prefix from the rule is treated specially, so “- foo + bar” is parsed as two rules (assuming that prefix-parsing wasn’t also disabled).
- o You may also specify any of the modifiers for the “+” or “-” rules (below) in order to have the rules that are read in from the file default to having that modifier set. For instance, “merge,-/.excl” would treat the contents of .excl as absolute-path excludes, while “dir-merge,s.filt” and

“:sC” would each make all their per-directory rules apply only on the sending side.

The following modifiers are accepted after a “+” or “-”:

- o A **/** specifies that the include/exclude rule should be matched against the absolute pathname of the current item. For example, “-/ /etc/passwd” would exclude the passwd file any time the transfer was sending files from the “/etc” directory, and “-/ subdir/foo” would always exclude “foo” when it is in a dir named “subdir”, even if “foo” is at the root of the current transfer.
- o A **!** specifies that the include/exclude should take effect if the pattern fails to match. For instance, “-! */” would exclude all non-directories.
- o A **C** is used to indicate that all the global CVS-exclude rules should be inserted as excludes in place of the “-C”. No arg should follow.
- o An **s** is used to indicate that the rule applies to the sending side. When a rule affects the sending side, it prevents files from being transferred. The default is for a rule to affect both sides unless **--delete-excluded** was specified, in which case default rules become sender-side only. See also the hide (H) and show (S) rules, which are an alternate way to specify sending-side includes/excludes.
- o An **r** is used to indicate that the rule applies to the receiving side. When a rule affects the receiving side, it prevents files from being deleted. See the **s** modifier for more info. See also the protect (P) and risk (R) rules, which are an alternate way to specify receiver-side includes/excludes.
- o A **p** indicates that a rule is perishable, meaning that it is ignored in directories that are being deleted. For instance, the **-C** option’s default rules that exclude things like “CVS” and “*.o” are marked as perishable, and will not prevent a directory that was removed on the source from being deleted on the destination.

Per-directory rules are inherited in all subdirectories of the directory where the merge-file was found unless the ‘n’ modifier was used. Each subdirectory’s rules are prefixed to the inherited per-directory rules from its parents, which gives the newest rules a higher priority than the inherited rules. The entire set of dir-merge rules are grouped together in the spot where the merge-file was specified, so it is possible to override dir-merge rules via a rule that got specified earlier in the list of global rules. When the list-clearing rule (“!”) is read from a per-directory file, it only clears the inherited rules for the current merge file.

Another way to prevent a single rule from a dir-merge file from being inherited is to anchor it with a leading slash. Anchored rules in a per-directory merge-file are relative to the merge-file’s directory, so a pattern “/foo” would only match the file “foo” in the directory where the dir-merge filter file was found.

Here’s an example filter file which you’d specify via **--filter=". file"**:

```
merge /home/user/.global-filter
- *.gz
dir-merge .rules
+ *.[ch]
- *.o
```

This will merge the contents of the /home/user/.global-filter file at the start of the list and also turns the “.rules” filename into a per-directory filter file. All rules read in prior to the start of the directory scan follow the global anchoring rules (i.e. a leading slash matches at the root of the transfer).

If a per-directory merge-file is specified with a path that is a parent directory of the first transfer directory, rsync will scan all the parent dirs from that starting point to the transfer directory for the indicated per-directory file. For instance, here is a common filter (see **-F**):

```
--filter=' : /.rsync-filter'
```

That rule tells rsync to scan for the file `.rsync-filter` in all directories from the root down through the parent directory of the transfer prior to the start of the normal directory scan of the file in the directories that are sent as a part of the transfer. (Note: for an rsync daemon, the root is always the same as the module's "path".)

Some examples of this pre-scanning for per-directory files:

```
rsync -avF /src/path/ /dest/dir
rsync -av --filter=': ../../.rsync-filter' /src/path/ /dest/dir
rsync -av --filter=': .rsync-filter' /src/path/ /dest/dir
```

The first two commands above will look for `.rsync-filter` in `/` and `/src` before the normal scan begins looking for the file in `/src/path` and its subdirectories. The last command avoids the parent-dir scan and only looks for the `.rsync-filter` files in each directory that is a part of the transfer.

If you want to include the contents of a `.cvsignore` in your patterns, you should use the rule `:C`, which creates a dir-merge of the `.cvsignore` file, but parsed in a CVS-compatible manner. You can use this to affect where the **--cvs-exclude (-C)** option's inclusion of the per-directory `.cvsignore` file gets placed into your rules by putting the `:C` wherever you like in your filter rules. Without this, rsync would add the dir-merge rule for the `.cvsignore` file at the end of all your other rules (giving it a lower priority than your command-line rules). For example:

```
cat <<EOT | rsync -avC --filter='.' -' a/ b
+ foo.o
:C
- *.old
EOT
rsync -avC --include=foo.o -f :C --exclude='*.old' a/ b
```

Both of the above rsync commands are identical. Each one will merge all the per-directory `.cvsignore` rules in the middle of the list rather than at the end. This allows their dir-specific rules to supersede the rules that follow the `:C` instead of being subservient to all your rules. To affect the other CVS exclude rules (i.e. the default list of exclusions, the contents of `$HOME/.cvsignore`, and the value of `$CVSIGNORE`) you should omit the `-C` command-line option and instead insert a `-C` rule into your filter rules; e.g. **--filter=-C**.

LIST-CLEARING FILTER RULE

You can clear the current include/exclude list by using the `!` filter rule (as introduced in the FILTER RULES section above). The "current" list is either the global list of rules (if the rule is encountered while parsing the filter options) or a set of per-directory rules (which are inherited in their own sub-list, so a sub-directory can use this to clear out the parent's rules).

ANCHORING INCLUDE/EXCLUDE PATTERNS

As mentioned earlier, global include/exclude patterns are anchored at the "root of the transfer" (as opposed to per-directory patterns, which are anchored at the merge-file's directory). If you think of the transfer as a subtree of names that are being sent from sender to receiver, the transfer-root is where the tree starts to be duplicated in the destination directory. This root governs where patterns that start with a `/` match.

Because the matching is relative to the transfer-root, changing the trailing slash on a source path or changing your use of the **--relative** option affects the path you need to use in your matching (in addition to changing how much of the file tree is duplicated on the destination host). The following examples demonstrate this.

Let's say that we want to match two source files, one with an absolute path of `/home/me/foo/bar`, and one with a path of `/home/you/bar/baz`. Here is how the various command choices differ for a 2-source transfer:

```
Example cmd: rsync -a /home/me /home/you /dest
+/- pattern: /me/foo/bar
+/- pattern: /you/bar/baz
```

Target file: /dest/me/foo/bar
 Target file: /dest/you/bar/baz

Example cmd: `rsync -a /home/me/ /home/you/ /dest`
 +/- pattern: /foo/bar (note missing “me”)
 +/- pattern: /bar/baz (note missing “you”)
 Target file: /dest/foo/bar
 Target file: /dest/bar/baz

Example cmd: `rsync -a --relative /home/me/ /home/you /dest`
 +/- pattern: /home/me/foo/bar (note full path)
 +/- pattern: /home/you/bar/baz (ditto)
 Target file: /dest/home/me/foo/bar
 Target file: /dest/home/you/bar/baz

Example cmd: `cd /home; rsync -a --relative me/foo you/ /dest`
 +/- pattern: /me/foo/bar (starts at specified path)
 +/- pattern: /you/bar/baz (ditto)
 Target file: /dest/me/foo/bar
 Target file: /dest/you/bar/baz

The easiest way to see what name you should filter is to just look at the output when using **--verbose** and put a / in front of the name (use the **--dry-run** option if you’re not yet ready to copy any files).

PER-DIRECTORY RULES AND DELETE

Without a delete option, per-directory rules are only relevant on the sending side, so you can feel free to exclude the merge files themselves without affecting the transfer. To make this easy, the ‘e’ modifier adds this exclude for you, as seen in these two equivalent commands:

```
rsync -av --filter=': .excl' --exclude=.excl host:src/dir /dest
rsync -av --filter=':e .excl' host:src/dir /dest
```

However, if you want to do a delete on the receiving side AND you want some files to be excluded from being deleted, you’ll need to be sure that the receiving side knows what files to exclude. The easiest way is to include the per-directory merge files in the transfer and use **--delete-after**, because this ensures that the receiving side gets all the same exclude rules as the sending side before it tries to delete anything:

```
rsync -avF --delete-after host:src/dir /dest
```

However, if the merge files are not a part of the transfer, you’ll need to either specify some global exclude rules (i.e. specified on the command line), or you’ll need to maintain your own per-directory merge files on the receiving side. An example of the first is this (assume that the remote .rules files exclude themselves):

```
rsync -av --filter=': .rules' --filter='./my/extra.rules'
--delete host:src/dir /dest
```

In the above example the extra.rules file can affect both sides of the transfer, but (on the sending side) the rules are subservient to the rules merged from the .rules files because they were specified after the per-directory merge rule.

In one final example, the remote side is excluding the .rsync-filter files from the transfer, but we want to use our own .rsync-filter files to control what gets deleted on the receiving side. To do this we must specifically exclude the per-directory merge files (so that they don’t get deleted) and then put rules into the local files to control what else should not get deleted. Like one of these commands:


```
rsync -av --filter=':e /rsync-filter' --delete \
  host:src/dir /dest
rsync -avFF --delete host:src/dir /dest
```

BATCH MODE

Batch mode can be used to apply the same set of updates to many identical systems. Suppose one has a tree which is replicated on a number of hosts. Now suppose some changes have been made to this source tree and those changes need to be propagated to the other hosts. In order to do this using batch mode, rsync is run with the write-batch option to apply the changes made to the source tree to one of the destination trees. The write-batch option causes the rsync client to store in a “batch file” all the information needed to repeat this operation against other, identical destination trees.

To apply the recorded changes to another destination tree, run rsync with the read-batch option, specifying the name of the same batch file, and the destination tree. Rsync updates the destination tree using the information stored in the batch file.

For convenience, one additional file is created when the write-batch option is used. This file’s name is created by appending “.sh” to the batch filename. The .sh file contains a command-line suitable for updating a destination tree using that batch file. It can be executed using a Bourne (or Bourne-like) shell, optionally passing in an alternate destination tree pathname which is then used instead of the original path. This is useful when the destination tree path differs from the original destination tree path.

Generating the batch file once saves having to perform the file status, checksum, and data block generation more than once when updating multiple destination trees. Multicast transport protocols can be used to transfer the batch update files in parallel to many hosts at once, instead of sending the same data to every host individually.

Examples:

```
$ rsync --write-batch=foo -a host:/source/dir/ /adest/dir/
$ scp foo* remote:
$ ssh remote ./foo.sh /bdest/dir/

$ rsync --write-batch=foo -a /source/dir/ /adest/dir/
$ ssh remote rsync --read-batch=- -a /bdest/dir/ <foo
```

In these examples, rsync is used to update /adest/dir/ from /source/dir/ and the information to repeat this operation is stored in “foo” and “foo.sh”. The host “remote” is then updated with the batched data going into the directory /bdest/dir. The differences between the two examples reveals some of the flexibility you have in how you deal with batches:

- o The first example shows that the initial copy doesn’t have to be local — you can push or pull data to/from a remote host using either the remote-shell syntax or rsync daemon syntax, as desired.
- o The first example uses the created “foo.sh” file to get the right rsync options when running the read-batch command on the remote host.
- o The second example reads the batch data via standard input so that the batch file doesn’t need to be copied to the remote machine first. This example avoids the foo.sh script because it needed to use a modified **--read-batch** option, but you could edit the script file if you wished to make use of it (just be sure that no other option is trying to use standard input, such as the **--exclude-from=-** option).

Caveats:

The read-batch option expects the destination tree that it is updating to be identical to the destination tree that was used to create the batch update fileset. When a difference between the destination trees is encountered the update might be discarded with a warning (if the file appears to be up-to-date already) or the file-

update may be attempted and then, if the file fails to verify, the update discarded with an error. This means that it should be safe to re-run a read-batch operation if the command got interrupted. If you wish to force the batched-update to always be attempted regardless of the file's size and date, use the **-I** option (when reading the batch). If an error occurs, the destination tree will probably be in a partially updated state. In that case, rsync can be used in its regular (non-batch) mode of operation to fix up the destination tree.

The rsync version used on all destinations must be at least as new as the one used to generate the batch file. Rsync will die with an error if the protocol version in the batch file is too new for the batch-reading rsync to handle. See also the **--protocol** option for a way to have the creating rsync generate a batch file that an older rsync can understand. (Note that batch files changed format in version 2.6.3, so mixing versions older than that with newer versions will not work.)

When reading a batch file, rsync will force the value of certain options to match the data in the batch file if you didn't set them to the same as the batch-writing command. Other options can (and should) be changed. For instance **--write-batch** changes to **--read-batch**, **--files-from** is dropped, and the **--filter/--include/--exclude** options are not needed unless one of the **--delete** options is specified.

The code that creates the BATCH.sh file transforms any filter/include/exclude options into a single list that is appended as a "here" document to the shell script file. An advanced user can use this to modify the exclude list if a change in what gets deleted by **--delete** is desired. A normal user can ignore this detail and just use the shell script as an easy way to run the appropriate **--read-batch** command for the batched data.

The original batch mode in rsync was based on "rsync+", but the latest version uses a new implementation.

SYMBOLIC LINKS

Three basic behaviors are possible when rsync encounters a symbolic link in the source directory.

By default, symbolic links are not transferred at all. A message "skipping non-regular" file is emitted for any symlinks that exist.

If **--links** is specified, then symlinks are recreated with the same target on the destination. Note that **--archive** implies **--links**.

If **--copy-links** is specified, then symlinks are "collapsed" by copying their referent, rather than the symlink.

rsync also distinguishes "safe" and "unsafe" symbolic links. An example where this might be used is a web site mirror that wishes ensure the rsync module they copy does not include symbolic links to **/etc/passwd** in the public section of the site. Using **--copy-unsafe-links** will cause any links to be copied as the file they point to on the destination. Using **--safe-links** will cause unsafe links to be omitted altogether. (Note that you must specify **--links** for **--safe-links** to have any effect.)

Symbolic links are considered unsafe if they are absolute symlinks (start with **/**), empty, or if they contain enough **".."** components to ascend from the directory being copied.

Here's a summary of how the symlink options are interpreted. The list is in order of precedence, so if your combination of options isn't mentioned, use the first line that is a complete subset of your options:

--copy-links

Turn all symlinks into normal files (leaving no symlinks for any other options to affect).

--links --copy-unsafe-links

Turn all unsafe symlinks into files and duplicate all safe symlinks.

--copy-unsafe-links

Turn all unsafe symlinks into files, noisily skip all safe symlinks.

--links --safe-links

Duplicate safe symlinks and skip unsafe ones.

--links

Duplicate all symlinks.

DIAGNOSTICS

rsync occasionally produces error messages that may seem a little cryptic. The one that seems to cause the most confusion is “protocol version mismatch — is your shell clean?”.

This message is usually caused by your startup scripts or remote shell facility producing unwanted garbage on the stream that rsync is using for its transport. The way to diagnose this problem is to run your remote shell like this:

```
ssh remotehost /bin/true > out.dat
```

then look at out.dat. If everything is working correctly then out.dat should be a zero length file. If you are getting the above error from rsync then you will probably find that out.dat contains some text or data. Look at the contents and try to work out what is producing it. The most common cause is incorrectly configured shell startup scripts (such as .cshrc or .profile) that contain output statements for non-interactive logins.

If you are having trouble debugging filter patterns, then try specifying the **-vv** option. At this level of verbosity rsync will show why each individual file is included or excluded.

EXIT VALUES

- | | |
|-----------|--|
| 0 | Success |
| 1 | Syntax or usage error |
| 2 | Protocol incompatibility |
| 3 | Errors selecting input/output files, dirs |
| 4 | Requested action not supported: an attempt was made to manipulate 64-bit files on a platform that cannot support them; or an option was specified that is supported by the client and not by the server. |
| 5 | Error starting client-server protocol |
| 6 | Daemon unable to append to log-file |
| 10 | Error in socket I/O |
| 11 | Error in file I/O |
| 12 | Error in rsync protocol data stream |
| 13 | Errors with program diagnostics |
| 14 | Error in IPC code |
| 20 | Received SIGUSR1 or SIGINT |
| 21 | Some error returned by <code>waitpid()</code> |
| 22 | Error allocating core memory buffers |
| 23 | Partial transfer due to error |
| 24 | Partial transfer due to vanished source files |
| 25 | The --max-delete limit stopped deletions |
| 30 | Timeout in data send/receive |
| 35 | Timeout waiting for daemon connection |

ENVIRONMENT VARIABLES

CVSIGNORE

The CVSIGNORE environment variable supplements any ignore patterns in .cvsignore files. See the **--cvs-exclude** option for more details.

RSYNC_ICONV

Specify a default **--iconv** setting using this environment variable.

RSYNC_RSH

The RSYNC_RSH environment variable allows you to override the default shell used as the transport for rsync. Command line options are permitted after the command name, just as in the **-e** option.

RSYNC_PROXY

The RSYNC_PROXY environment variable allows you to redirect your rsync client to use a web proxy when connecting to a rsync daemon. You should set RSYNC_PROXY to a hostname:port pair.

RSYNC_PASSWORD

Setting RSYNC_PASSWORD to the required password allows you to run authenticated rsync connections to an rsync daemon without user intervention. Note that this does not supply a password to a remote shell transport such as ssh; to learn how to do that, consult the remote shell's documentation.

USER or LOGNAME

The USER or LOGNAME environment variables are used to determine the default username sent to an rsync daemon. If neither is set, the username defaults to "nobody".

HOME

The HOME environment variable is used to find the user's default .cvsignore file.

FILES

/etc/rsyncd.conf or rsyncd.conf

SEE ALSO

rsyncd.conf(5)

BUGS

times are transferred as *nix time_t values

When transferring to FAT filesystems rsync may re-sync unmodified files. See the comments on the **--modify-window** option.

file permissions, devices, etc. are transferred as native numerical values

see also the comments on the **--delete** option

Please report bugs! See the web site at <http://rsync.samba.org/>

VERSION

This man page is current for version 3.0.2 of rsync.

INTERNAL OPTIONS

The options **--server** and **--sender** are used internally by rsync, and should never be typed by a user under normal circumstances. Some awareness of these options may be needed in certain scenarios, such as when setting up a login that can only run an rsync command. For instance, the support directory of the rsync distribution has an example script named rrsync (for restricted rsync) that can be used with a restricted ssh login.

CREDITS

rsync is distributed under the GNU public license. See the file COPYING for details.

A WEB site is available at <http://rsync.samba.org/>. The site includes an FAQ-O-Matic which may cover questions unanswered by this manual page.

The primary ftp site for rsync is <ftp://rsync.samba.org/pub/rsync>.

We would be delighted to hear from you if you like this program. Please contact the mailing-list at rsync@lists.samba.org.

This program uses the excellent zlib compression library written by Jean-loup Gailly and Mark Adler.

THANKS

Especial thanks go out to: John Van Essen, Matt McCutchen, Wesley W. Terpstra, David Dykstra, Jos Backus, Sebastian Krahmer, Martin Pool, and our gone-but-not-forgotten compadre, J.W. Schultz.

Thanks also to Richard Brent, Brendan Mackay, Bill Waite, Stephen Rothwell and David Bell. I've probably missed some people, my apologies if I have.

AUTHOR

rsync was originally written by Andrew Tridgell and Paul Mackerras. Many people have later contributed to it. It is currently maintained by Wayne Davison.

Mailing lists for support and development are available at <http://lists.samba.org>

NAME

rsyncd.conf — configuration file for rsync in daemon mode

SYNOPSIS

rsyncd.conf

DESCRIPTION

The rsyncd.conf file is the runtime configuration file for rsync when run as an rsync daemon.

The rsyncd.conf file controls authentication, access, logging and available modules.

FILE FORMAT

The file consists of modules and parameters. A module begins with the name of the module in square brackets and continues until the next module begins. Modules contain parameters of the form “name = value”.

The file is line-based — that is, each newline-terminated line represents either a comment, a module name or a parameter.

Only the first equals sign in a parameter is significant. Whitespace before or after the first equals sign is discarded. Leading, trailing and internal whitespace in module and parameter names is irrelevant. Leading and trailing whitespace in a parameter value is discarded. Internal whitespace within a parameter value is retained verbatim.

Any line beginning with a hash (#) is ignored, as are lines containing only whitespace.

Any line ending in a \ is “continued” on the next line in the customary UNIX fashion.

The values following the equals sign in parameters are all either a string (no quotes needed) or a boolean, which may be given as yes/no, 0/1 or true/false. Case is not significant in boolean values, but is preserved in string values.

LAUNCHING THE RSYNC DAEMON

The rsync daemon is launched by specifying the **--daemon** option to rsync.

The daemon must run with root privileges if you wish to use chroot, to bind to a port numbered under 1024 (as is the default 873), or to set file ownership. Otherwise, it must just have permission to read and write the appropriate data, log, and lock files.

You can launch it either via inetd, as a stand-alone daemon, or from an rsync client via a remote shell. If run as a stand-alone daemon then just run the command “**rsync --daemon**” from a suitable startup script.

When run via inetd you should add a line like this to /etc/services:

```
rsync      873/tcp
```

and a single line something like this to /etc/inetd.conf:

```
rsync  stream tcp  nowait root  /usr/bin/rsync rsyncd --daemon
```

Replace “/usr/bin/rsync” with the path to where you have rsync installed on your system. You will then need to send inetd a HUP signal to tell it to reread its config file.

Note that you should **not** send the rsync daemon a HUP signal to force it to reread the `rsyncd.conf` file. The file is re-read on each client connection.

GLOBAL OPTIONS

The first parameters in the file (before a [module] header) are the global parameters.

You may also include any module parameters in the global part of the config file in which case the supplied value will override the default for that parameter.

motd file

The “motd file” option allows you to specify a “message of the day” to display to clients on each connect. This usually contains site information and any legal notices. The default is no motd file.

pid file The “pid file” option tells the rsync daemon to write its process ID to that file. If the file already exists, the rsync daemon will abort rather than overwrite the file.

port You can override the default port the daemon will listen on by specifying this value (defaults to 873). This is ignored if the daemon is being run by inetd, and is superseded by the **--port** command-line option.

address

You can override the default IP address the daemon will listen on by specifying this value. This is ignored if the daemon is being run by inetd, and is superseded by the **--address** command-line option.

socket options

This option can provide endless fun for people who like to tune their systems to the utmost degree. You can set all sorts of socket options which may make transfers faster (or slower!). Read the man page for the `setsockopt()` system call for details on some of the options you may be able to set. By default no special socket options are set. These settings are superseded by the **--sockopts** command-line option.

slp refresh

This option is used to determine how long service advertisements are valid (measured in seconds), and is only applicable if you have Service Location Protocol support compiled in. If this option is not set or is set to zero, then service advertisements never time out. If this is set to less than 120 seconds, then 120 seconds is used. If it is set to more than 65535, then 65535 is used (which is a limitation of SLP). Using 3600 (one hour) is a good number if you tend to change your configuration.

MODULE OPTIONS

After the global options you should define a number of modules, each module exports a directory tree as a symbolic name. Modules are exported by specifying a module name in square brackets [module] followed by the options for that module. The module name cannot contain a slash or a closing square bracket. If the name contains whitespace, each internal sequence of whitespace will be changed into a single space, while leading or trailing whitespace will be discarded.

comment

The “comment” option specifies a description string that is displayed next to the module name when clients obtain a list of available modules. The default is no comment.

path The “path” option specifies the directory in the daemon’s filesystem to make available in this module. You must specify this option for each module in `rsyncd.conf`.

use chroot

If “use chroot” is true, the rsync daemon will chroot to the “path” before starting the file transfer with the client. This has the advantage of extra protection against possible implementation security holes, but it has the disadvantages of requiring super-user privileges, of not being able to follow symbolic links that are either absolute or outside of the new root path, and of complicating the preservation of users and groups by name (see below).

As an additional safety feature, you can specify a dot-dir in the module’s “path” to indicate the point where the chroot should occur. This allows rsync to run in a chroot with a non-“/” path for the top of the transfer hierarchy. Doing this guards against unintended library loading (since those absolute paths will not be inside the transfer hierarchy unless you have used an unwise pathname), and lets you setup libraries for the chroot that are outside of the transfer. For example, specifying “/var/rsync/.module1” will chroot to the “/var/rsync” directory and set the inside-chroot path to “/module1”. If you had omitted the dot-dir, the chroot would have used the whole path, and the inside-chroot path would have been “/”.

When “use chroot” is false or the inside-chroot path is not “/”, rsync will: (1) munge symlinks by default for security reasons (see “munge symlinks” for a way to turn this off, but only if you trust your users), (2) substitute leading slashes in absolute paths with the module’s path (so that options

such as **—backup-dir**, **—compare-dest**, etc. interpret an absolute path as rooted in the module’s “path” dir), and (3) trim “..” path elements from args if rsync believes they would escape the module hierarchy. The default for “use chroot” is true, and is the safer choice (especially if the module is not read-only).

When this option is enabled, rsync will not attempt to map users and groups by name (by default), but instead copy IDs as though **—numeric-ids** had been specified. In order to enable name-mapping, rsync needs to be able to use the standard library functions for looking up names and IDs (i.e. `getpwuid()`, `getgrgid()`, `getpwnam()`, and `getgrnam()`). This means the rsync process in the chroot hierarchy will need to have access to the resources used by these library functions (traditionally `/etc/passwd` and `/etc/group`, but perhaps additional dynamic libraries as well).

If you copy the necessary resources into the module’s chroot area, you should protect them through your OS’s normal user/group or ACL settings (to prevent the rsync module’s user from being able to change them), and then hide them from the user’s view via “exclude” (see how in the discussion of that option). At that point it will be safe to enable the mapping of users and groups by name using the “numeric ids” daemon option (see below).

Note also that you are free to setup custom user/group information in the chroot area that is different from your normal system. For example, you could abbreviate the list of users and groups.

numeric ids

Enabling the “numeric ids” option disables the mapping of users and groups by name for the current daemon module. This prevents the daemon from trying to load any user/group-related files or libraries. Enabling this option makes the transfer behave as if the client had passed the **—numeric-ids** command-line option. By default, this parameter is enabled for chroot modules and disabled for non-chroot modules.

A chroot-enabled module should not have this option enabled unless you’ve taken steps to ensure that the module has the necessary resources it needs to translate names, and that it is not possible for a user to change those resources.

munge symlinks

The “munge symlinks” option tells rsync to modify all incoming symlinks in a way that makes them unusable but recoverable (see below). This should help protect your files from user trickery when your daemon module is writable. The default is disabled when “use chroot” is on and the inside-chroot path is “/”, otherwise it is enabled.

If you disable this option on a daemon that is not read-only, there are tricks that a user can play with uploaded symlinks to access daemon-excluded items (if your module has any), and, if “use chroot” is off, rsync can even be tricked into showing or changing data that is outside the module’s path (as access-permissions allow).

The way rsync disables the use of symlinks is to prefix each one with the string “/rsyncd-munged/”. This prevents the links from being used as long as that directory does not exist. When this option is enabled, rsync will refuse to run if that path is a directory or a symlink to a directory. When using the “munge symlinks” option in a chroot area that has an inside-chroot path of “/”, you should add “/rsyncd-munged/” to the exclude setting for the module so that a user can’t try to create it.

Note: rsync makes no attempt to verify that any pre-existing symlinks in the hierarchy are as safe as you want them to be. If you setup an rsync daemon on a new area or locally add symlinks, you can manually protect your symlinks from being abused by prefixing “/rsyncd-munged/” to the start of every symlink’s value. There is a perl script in the support directory of the source code named “munge-symlinks” that can be used to add or remove this prefix from your symlinks.

When this option is disabled on a writable module and “use chroot” is off (or the inside-chroot path is not “/”), incoming symlinks will be modified to drop a leading slash and to remove “..” path elements that rsync believes will allow a symlink to escape the module’s hierarchy. There are

tricky ways to work around this, though, so you had better trust your users if you choose this combination of options.

charset

This specifies the name of the character set in which the module's filenames are stored. If the client uses an **--iconv** option, the daemon will use the value of the "charset" parameter regardless of the character set the client actually passed. This allows the daemon to support charset conversion in a chroot module without extra files in the chroot area, and also ensures that name-translation is done in a consistent manner. If the "charset" parameter is not set, the **--iconv** option is refused, just as if "iconv" had been specified via "refuse options".

If you wish to force users to always use **--iconv** for a particular module, add "no-iconv" to the "refuse options" parameter. Keep in mind that this will restrict access to your module to very new rsync clients.

max connections

The "max connections" option allows you to specify the maximum number of simultaneous connections you will allow. Any clients connecting when the maximum has been reached will receive a message telling them to try later. The default is 0, which means no limit. A negative value disables the module. See also the "lock file" option.

log file When the "log file" option is set to a non-empty string, the rsync daemon will log messages to the indicated file rather than using syslog. This is particularly useful on systems (such as AIX) where `syslog()` doesn't work for chrooted programs. The file is opened before `chroot()` is called, allowing it to be placed outside the transfer. If this value is set on a per-module basis instead of globally, the global log will still contain any authorization failures or config-file error messages.

If the daemon fails to open to specified file, it will fall back to using syslog and output an error about the failure. (Note that the failure to open the specified log file used to be a fatal error.)

syslog facility

The "syslog facility" option allows you to specify the syslog facility name to use when logging messages from the rsync daemon. You may use any standard syslog facility name which is defined on your system. Common names are auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, syslog, user, uucp, local0, local1, local2, local3, local4, local5, local6 and local7. The default is daemon. This setting has no effect if the "log file" setting is a non-empty string (either set in the per-modules settings, or inherited from the global settings).

max verbosity

The "max verbosity" option allows you to control the maximum amount of verbose information that you'll allow the daemon to generate (since the information goes into the log file). The default is 1, which allows the client to request one level of verbosity.

lock file

The "lock file" option specifies the file to use to support the "max connections" option. The rsync daemon uses record locking on this file to ensure that the max connections limit is not exceeded for the modules sharing the lock file. The default is `/var/run/rsyncd.lock`.

read only

The "read only" option determines whether clients will be able to upload files or not. If "read only" is true then any attempted uploads will fail. If "read only" is false then uploads will be possible if file permissions on the daemon side allow them. The default is for all modules to be read only.

write only

The "write only" option determines whether clients will be able to download files or not. If "write only" is true then any attempted downloads will fail. If "write only" is false then downloads will be possible if file permissions on the daemon side allow them. The default is for this option to be disabled.

- list** The “list” option determines if this module should be listed when the client asks for a listing of available modules. By setting this to false you can create hidden modules. The default is for modules to be listable.
- uid** The “uid” option specifies the user name or user ID that file transfers to and from that module should take place as when the daemon was run as root. In combination with the “gid” option this determines what file permissions are available. The default is uid -2, which is normally the user “nobody”.
- gid** The “gid” option specifies the group name or group ID that file transfers to and from that module should take place as when the daemon was run as root. This complements the “uid” option. The default is gid -2, which is normally the group “nobody”.

fake super

Setting “fake super = yes” for a module causes the daemon side to behave as if the **—fake-user** command-line option had been specified. This allows the full attributes of a file to be stored without having to have the daemon actually running as root.

- filter** The daemon has its own filter chain that determines what files it will let the client access. This chain is not sent to the client and is independent of any filters the client may have specified. Files excluded by the daemon filter chain (**daemon-excluded** files) are treated as non-existent if the client tries to pull them, are skipped with an error message if the client tries to push them (triggering exit code 23), and are never deleted from the module. You can use daemon filters to prevent clients from downloading or tampering with private administrative files, such as files you may add to support uid/gid name translations.

The daemon filter chain is built from the “filter”, “include from”, “include”, “exclude from”, and “exclude” parameters, in that order of priority. Anchored patterns are anchored at the root of the module. To prevent access to an entire subtree, for example, “/secret”, you *must* exclude everything in the subtree; the easiest way to do this is with a triple-star pattern like “/secret/**”.

The “filter” parameter takes a space-separated list of daemon filter rules, though it is smart enough to know not to split a token at an internal space in a rule (e.g. “- /foo — /bar” is parsed as two rules). You may specify one or more merge-file rules using the normal syntax. Only one “filter” parameter can apply to a given module in the config file, so put all the rules you want in a single parameter. Note that per-directory merge-file rules do not provide as much protection as global rules, but they can be used to make **—delete** work better during a client download operation if the per-dir merge files are included in the transfer and the client requests that they be used.

exclude

The “exclude” parameter takes a space-separated list of daemon exclude patterns. As with the client **—exclude** option, patterns can be qualified with “- ” or “+ ” to explicitly indicate exclude/include. Only one “exclude” parameter can apply to a given module. See the “filter” parameter for a description of how excluded files affect the daemon.

include

Use an “include” to override the effects of the “exclude” parameter. Only one “include” parameter can apply to a given module. See the “filter” parameter for a description of how excluded files affect the daemon.

exclude from

The “exclude from” parameter specifies the name of a file on the daemon that contains daemon exclude patterns, one per line. Only one “exclude from” parameter can apply to a given module; if you have multiple exclude-from files, you can specify them as a merge file in the “filter” parameter. See the “filter” parameter for a description of how excluded files affect the daemon.

include from

Analogue of “exclude from” for a file of daemon include patterns. Only one “include from” parameter can apply to a given module. See the “filter” parameter for a description of how excluded files affect the daemon.

incoming chmod

This option allows you to specify a set of comma-separated chmod strings that will affect the permissions of all incoming files (files that are being received by the daemon). These changes happen after all other permission calculations, and this will even override destination-default and/or existing permissions when the client does not specify **--perms**. See the description of the **--chmod** rsync option and the **chmod(1)** manpage for information on the format of this string.

outgoing chmod

This option allows you to specify a set of comma-separated chmod strings that will affect the permissions of all outgoing files (files that are being sent out from the daemon). These changes happen first, making the sent permissions appear to be different than those stored in the filesystem itself. For instance, you could disable group write permissions on the server while having it appear to be on to the clients. See the description of the **--chmod** rsync option and the **chmod(1)** manpage for information on the format of this string.

auth users

The “auth users” option specifies a comma and space-separated list of usernames that will be allowed to connect to this module. The usernames do not need to exist on the local system. The usernames may also contain shell wildcard characters. If “auth users” is set then the client will be challenged to supply a username and password to connect to the module. A challenge response authentication protocol is used for this exchange. The plain text usernames and passwords are stored in the file specified by the “secrets file” option. The default is for all users to be able to connect without a password (this is called “anonymous rsync”).

See also the “CONNECTING TO AN RSYNC DAEMON OVER A REMOTE SHELL PROGRAM” section in **rsync(1)** for information on how handle an rsyncd.conf–level username that differs from the remote-shell-level username when using a remote shell to connect to an rsync daemon.

secrets file

The “secrets file” option specifies the name of a file that contains the username:password pairs used for authenticating this module. This file is only consulted if the “auth users” option is specified. The file is line based and contains username:password pairs separated by a single colon. Any line starting with a hash (#) is considered a comment and is skipped. The passwords can contain any characters but be warned that many operating systems limit the length of passwords that can be typed at the client end, so you may find that passwords longer than 8 characters don’t work.

There is no default for the “secrets file” option, you must choose a name (such as `/etc/rsyncd.secrets`). The file must normally not be readable by “other”; see “strict modes”.

strict modes

The “strict modes” option determines whether or not the permissions on the secrets file will be checked. If “strict modes” is true, then the secrets file must not be readable by any user ID other than the one that the rsync daemon is running under. If “strict modes” is false, the check is not performed. The default is true. This option was added to accommodate rsync running on the Windows operating system.

hosts allow

The “hosts allow” option allows you to specify a list of patterns that are matched against a connecting clients hostname and IP address. If none of the patterns match then the connection is rejected.

Each pattern can be in one of five forms:

- o a dotted decimal IPv4 address of the form a.b.c.d, or an IPv6 address of the form a:b:c::d:e:f. In this case the incoming machine’s IP address must match exactly.
- o an address/mask in the form ipaddr/n where ipaddr is the IP address and n is the number of one bits in the netmask. All IP addresses which match the masked IP address will be

allowed in.

- o an address/mask in the form `ipaddr/maskaddr` where `ipaddr` is the IP address and `maskaddr` is the netmask in dotted decimal notation for IPv4, or similar for IPv6, e.g. `ffff:ffff:ffff:ffff::` instead of `/64`. All IP addresses which match the masked IP address will be allowed in.
- o a hostname. The hostname as determined by a reverse lookup will be matched (case insensitive) against the pattern. Only an exact match is allowed in.
- o a hostname pattern using wildcards. These are matched using the same rules as normal unix filename matching. If the pattern matches then the client is allowed in.

Note IPv6 link-local addresses can have a scope in the address specification:

```
fe80::1%link1
fe80::%link1/64
fe80::%link1/ffff:ffff:ffff:ffff::
```

You can also combine “hosts allow” with a separate “hosts deny” option. If both options are specified then the “hosts allow” option is checked first and a match results in the client being able to connect. The “hosts deny” option is then checked and a match means that the host is rejected. If the host does not match either the “hosts allow” or the “hosts deny” patterns then it is allowed to connect.

The default is no “hosts allow” option, which means all hosts can connect.

hosts deny

The “hosts deny” option allows you to specify a list of patterns that are matched against a connecting clients hostname and IP address. If the pattern matches then the connection is rejected. See the “hosts allow” option for more information.

The default is no “hosts deny” option, which means all hosts can connect.

ignore errors

The “ignore errors” option tells rsync to ignore I/O errors on the daemon when deciding whether to run the delete phase of the transfer. Normally rsync skips the **—delete** step if any I/O errors have occurred in order to prevent disastrous deletion due to a temporary resource shortage or other I/O error. In some cases this test is counter productive so you can use this option to turn off this behavior.

ignore nonreadable

This tells the rsync daemon to completely ignore files that are not readable by the user. This is useful for public archives that may have some non-readable files among the directories, and the sysadmin doesn’t want those files to be seen at all.

transfer logging

The “transfer logging” option enables per-file logging of downloads and uploads in a format somewhat similar to that used by ftp daemons. The daemon always logs the transfer at the end, so if a transfer is aborted, no mention will be made in the log file.

If you want to customize the log lines, see the “log format” option.

log format

The “log format” option allows you to specify the format used for logging file transfers when transfer logging is enabled. The format is a text string containing embedded single-character escape sequences prefixed with a percent (%) character. An optional numeric field width may also be specified between the percent and the escape letter (e.g. “**%-50n %8l %07p**”).

The default log format is “**%o %h [%a] %m (%u) %f %l**”, and a “**%t [%p]**” is always prefixed when using the “log file” option. (A perl script that will summarize this default log format is

included in the rsync source code distribution in the “support” subdirectory: rsyncstats.)

The single-character escapes that are understood are as follows:

- o %a the remote IP address
- o %b the number of bytes actually transferred
- o %B the permission bits of the file (e.g. rwxrwxrwt)
- o %c the checksum bytes received for this file (only when sending)
- o %f the filename (long form on sender; no trailing “/”)
- o %G the gid of the file (decimal) or “DEFAULT”
- o %h the remote host name
- o %i an itemized list of what is being updated
- o %l the length of the file in bytes
- o %L the string “ -> SYMLINK”, “ => HARDLINK”, or “” (where **SYMLINK** or **HARDLINK** is a filename)
- o %m the module name
- o %M the last-modified time of the file
- o %n the filename (short form; trailing “/” on dir)
- o %o the operation, which is “send”, “recv”, or “del.” (the latter includes the trailing period)
- o %p the process ID of this rsync session
- o %P the module path
- o %t the current date time
- o %u the authenticated username or an empty string
- o %U the uid of the file (decimal)

For a list of what the characters mean that are output by “%i”, see the **--itemize-changes** option in the rsync manpage.

Note that some of the logged output changes when talking with older rsync versions. For instance, deleted files were only output as verbose messages prior to rsync 2.6.4.

timeout

The “timeout” option allows you to override the clients choice for I/O timeout for this module. Using this option you can ensure that rsync won’t wait on a dead client forever. The timeout is specified in seconds. A value of zero means no timeout and is the default. A good choice for anonymous rsync daemons may be 600 (giving a 10 minute timeout).

refuse options

The “refuse options” option allows you to specify a space-separated list of rsync command line options that will be refused by your rsync daemon. You may specify the full option name, its one-letter abbreviation, or a wild-card string that matches multiple options. For example, this would refuse **--checksum** (**-c**) and all the various delete options:

```
refuse options = c delete
```

The reason the above refuses all delete options is that the options imply **--delete**, and implied options are refused just like explicit options. As an additional safety feature, the refusal of “delete” also refuses **remove-source-files** when the daemon is the sender; if you want the latter without the former, instead refuse “delete-*” — that refuses all the delete modes without affecting

--remove-source-files.

When an option is refused, the daemon prints an error message and exits. To prevent all compression when serving files, you can use “dont compress = *” (see below) instead of “refuse options = compress” to avoid returning an error to a client that requests compression.

dont compress

The “dont compress” option allows you to select filenames based on wildcard patterns that should not be compressed when pulling files from the daemon (no analogous option exists to govern the pushing of files to a daemon). Compression is expensive in terms of CPU usage, so it is usually good to not try to compress files that won’t compress well, such as already compressed files.

The “dont compress” option takes a space-separated list of case-insensitive wildcard patterns. Any source filename matching one of the patterns will not be compressed during transfer.

See the **--skip-compress** option in the **rsync(1)** manpage for the list of file suffixes that are not compressed by default. Specifying a value for the “dont compress” option changes the default when the daemon is the sender.

pre-xfer exec, post-xfer exec

You may specify a command to be run before and/or after the transfer. If the **pre-xfer exec** command fails, the transfer is aborted before it begins.

The following environment variables will be set, though some are specific to the pre-xfer or the post-xfer environment:

- o **RSYNC_MODULE_NAME**: The name of the module being accessed.
- o **RSYNC_MODULE_PATH**: The path configured for the module.
- o **RSYNC_HOST_ADDR**: The accessing host’s IP address.
- o **RSYNC_HOST_NAME**: The accessing host’s name.
- o **RSYNC_USER_NAME**: The accessing user’s name (empty if no user).
- o **RSYNC_PID**: A unique number for this transfer.
- o **RSYNC_REQUEST**: (pre-xfer only) The module/path info specified by the user (note that the user can specify multiple source files, so the request can be something like “mod/path1 mod/path2”, etc.).
- o **RSYNC_ARG#**: (pre-xfer only) The pre-request arguments are set in these numbered values. **RSYNC_ARG0** is always “rsyncd”, and the last value contains a single period.
- o **RSYNC_EXIT_STATUS**: (post-xfer only) the server side’s exit value. This will be 0 for a successful run, a positive value for an error that the server generated, or a -1 if rsync failed to exit properly. Note that an error that occurs on the client side does not currently get sent to the server side, so this is not the final exit status for the whole transfer.
- o **RSYNC_RAW_STATUS**: (post-xfer only) the raw exit value from `waitpid()`.

Even though the commands can be associated with a particular module, they are run using the permissions of the user that started the daemon (not the module’s uid/gid setting) without any chroot restrictions.

AUTHENTICATION STRENGTH

The authentication protocol used in rsync is a 128 bit MD4 based challenge response system. This is fairly weak protection, though (with at least one brute-force hash-finding algorithm publicly available), so if you want really top-quality security, then I recommend that you run rsync over ssh. (Yes, a future version of rsync will switch over to a stronger hashing method.)

Also note that the rsync daemon protocol does not currently provide any encryption of the data that is transferred over the connection. Only authentication is provided. Use ssh as the transport if you want encryption.

Future versions of rsync may support SSL for better authentication and encryption, but that is still being investigated.

EXAMPLES

A simple rsyncd.conf file that allow anonymous rsync to a ftp area at `/home/ftp` would be:

```
[ftp]
  path = /home/ftp
  comment = ftp export area
```

A more sophisticated example would be:

```
uid = nobody
gid = nobody
use chroot = yes
max connections = 4
syslog facility = local5
pid file = /var/run/rsyncd.pid
slp refresh = 3600
```

```
[ftp]
  path = /var/ftp/.pub
  comment = whole ftp area (approx 6.1 GB)
```

```
[smbaftp]
  path = /var/ftp/.pub/samba
  comment = Samba ftp area (approx 300 MB)
```

```
[rsyncftp]
  path = /var/ftp/.pub/rsync
  comment = rsync ftp area (approx 6 MB)
```

```
[sambawww]
  path = /public_html/samba
  comment = Samba WWW pages (approx 240 MB)
```

```
[cvs]
  path = /data/cvs
  comment = CVS repository (requires authentication)
  auth users = tridge, susan
  secrets file = /etc/rsyncd.secrets
```

The `/etc/rsyncd.secrets` file would look something like this:

```
tridge:mypass
susan:herpass
```

FILES

`/etc/rsyncd.conf` or `rsyncd.conf`

SEE ALSO

rsync(1)

DIAGNOSTICS**BUGS**

Please report bugs! The rsync bug tracking system is online at <http://rsync.samba.org/>

VERSION

This man page is current for version 3.0.2 of rsync.

CREDITS

rsync is distributed under the GNU public license. See the file COPYING for details.

The primary ftp site for rsync is <ftp://rsync.samba.org/pub/rsync>.

A WEB site is available at <http://rsync.samba.org/>

We would be delighted to hear from you if you like this program.

This program uses the zlib compression library written by Jean-loup Gailly and Mark Adler.

THANKS

Thanks to Warren Stanley for his original idea and patch for the rsync daemon. Thanks to Karsten Thygesen for his many suggestions and documentation!

AUTHOR

rsync was written by Andrew Tridgell and Paul Mackerras. Many people have later contributed to it.

Mailing lists for support and development are available at <http://lists.samba.org>

NAME

sat – start edrc with a special System Administration configuration

SYNOPSIS

edrc/bin/sat [**-h** | **-V**]

sat [**-s**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Short start of **edrc** with an other configuration which points to an own script tree. Internally **sat** calls **edrc -c edrc.sat.cfg -n SAT_@ID@** .

sat stands for "System Administration Tool".

The session name (as shown in the menu) is automatically set to **SAT_<id>** . Where **<id>** is the process id of the started **edrc** instance if not already a session with the same name exists, if so the **<id>** is set to a random number.

If additional short starts are needed, create a symlink from the new short start command to **sat** (see section **EXAMPLES**).

OPTIONS

- h** usage message.
- V** print version and patch level of **edrc**. For an explanation of the release numbering system see **edrcrevision(1)**.
- s** silent startup. Startup without showing the EDRC banner.
- t** no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

ENVIRONMENT

-

EXIT STATUS

see **edrc(1m)**.

FILES

etc/edrc.sat.cfg

configuration file of **sat**, see **edrc.cfg(4)** for more information.

Other files see section **FILES** in **edrc(1m)**.

EXAMPLES**1) create a new short start**

This creates a new short start command **new_shortstart** that will load the configuration file **edrc.new_shortstart.cfg**. The session name will automatically be set to **NEW_SHORTSTART_<id>** .

```
[ /root ]
[ root@rh7mzv7t001 ] [bash]: cd ~edrc/bin

[ /opt/edrc/bin ]
[ root@acme007 ] [bash]: ln -s sat new_shortstart
```

or:

```
[ /opt/edrc/bin ]
[ root@acme007 ] [bash]: symlink sat ~edrc/bin/new_shortstart
```

SEE ALSO

edrc(1m), **edrc.cfg(4)**, **edrcintro(1)**, **edrcrevision(1)**, **symlink(1)**

NOTES

-

BUGS

-

AUTHOR

sat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sav – Make Backup Copy

SYNOPSIS

edrc/bin/sav -h

sav [**-q**] *file*...

sav [**-q**] { **-d** | **-D** *subdir* } *file*...

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **sav** utility is used by **mkuser** (for example) to make a backup copy of one or more files. Without options, the backup file will be named **file.\$TODAY** and stored in the same directory as the original. With a **-d** or **-D** *subdir* option, the backup file will be stored in a subdirectory.

Saves of files located in special operating system configuration or state directories are always done into a **.\$TODAY/** directory to ensure proper system functionality. See section **FILES** for a complete list of those directories.

OPTIONS

-q Quiet mode, does not give any warnings to standard output.

-d For each *file*, copy it to **.\$TODAY/file**, where TODAY is today's date in the form YYYYMMDD. If the directory does not exist, it is created.

-D *subdir* For each file, *file* copy it to **subdir.\$TODAY/file**, where TODAY is today's date in the form YYYYMMDD. If the directory does not exist, it is created. (This option is used by **mkuser**.)

ENVIRONMENT

\$TZ timezone setting.

EXIT STATUS

0 no error

4 usage displayed

8 could not create subdir or command interrupted

FILES

Files located in the following operating system configuration or state directories are always saved into the **.\$TODAY/** directory as if **sav** would be invoked using the **-d** option:

/etc/*.d/, /etc/*/*.d/, /etc/*/*/*.d, /etc/default/

as links to the rc scripts in **/etc/init.d** and configuration of rc scripts, as seen on HPUX, Linux and Solaris.

/sbin/*.d/ as links to the rc scripts in **/etc/init.d**, as seen on HPUX.

/var/spool/at/

defined at jobs, as seen on Linux.

/var/spool/cron/atjobs/

defined at jobs, as seen on HPUX and Solaris.

/var/spool/cron/

defined crontabs, as seen on Linux.

/var/spool/cron/crontabs/

defined crontabs, as seen on HPUX and Solaris.

***/man/man*/, */man/*/*man*/**

manual pages directories.

/lib/sypoll/protocols/

protocol plugins of **sypoll(1)**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **at(1)**, **cron(1m)**, **crontab(1m)**, **mkuser(1)**, **name(1)**, **timezone(3)**, **vsav(1)**

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

BUGS

-

AUTHOR

sav was developed by Peter Stevens, SFI and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

schedule.dat – schedule definition for lots

SYNOPSIS

edrc/var/lots/objects/schedule.dat
VARDIR/objects/schedule.dat

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the scheduling definition for the **lots** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

DATALIST ; DESCRIPTION ; HOSTNAME ; RETENTION ; COMPRESSION ; SCHEDULE ;

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS*DATALIST*

name of the data list as defined in the **datalist.dat** file. For each scheduled *DATALIST* in the **schedule.dat** file at least a *DATALIST* in the **datalist.dat** file must exist.

DESCRIPTION

description of the schedule.

Be aware that this should be a meaningful description due to the fact that this data is saved for a long time and people having done the definition might have moved on before the data might be needed for restore.

HOSTNAME

hostname where the schedule should be active.

RETENTION

number of days the data is locked for. If multiple schedules match for a certain day, the schedule having the highest retention will be executed.

If the defined *RETENTION* is outside the **RETENTION_MIN** and **RETENTION_MAX** range as defined in the **lots.cfg** file, the effective *RETENTION* is adjusted accordingly.

COMPRESSION

compression mode:

NONE

no compression.

GZIP

use **gzip** to compress the data. The command executed is: **gzip -cv --fast**.

COMPRESS

use **compress** to compress the data. The command executed is: **compress -cv**.

ZIP

use **zip** to compress the data. The command executed is: **zip**.

BZIP2

use **bzip2** to compress the data. The command executed is: **bzip2 -cv --fast**.

SCHEDULE

schedule definitions. This is a space separated list of schedule definitions of the format explained below. A schedule can consist on any combination of the following entries:

DD.

day of the month. This schedule will match each month on the day specified. Therefore the schedule **28.** will match each month at the 28th.

DD.MM.

specify day and month. This schedule will match each day of the month every year. Therefore the schedule **04.07.** will match on the July 4th every year.

DD.MM.YYYY

precise date specification. This schedule will only match on the precise date. Therefore the schedule **28.01.1986** will only match on the January 28th 1986. It is allowed to specify the schedule entry **00.00.0000** to mark a prepared but not really planned schedule.

WDay

short English weekday name: **Mon Tue Wed Thu Fri Sat Sun**. This schedule will only match on the weekday specified. Therefore the schedule **Fri** will match every Friday.

EXAMPLES

```
#
# var/lots/objects/schedule.dat - schedule and retention information
#
# [00] 30.09.2009 CWa Initial Version
#
# LEVEL          RETENTION          BACKUP (= DB export)    => SAVE_SCHEDULIN
```



```

# -----
# Silver      5 Months => 150 days  Sunday                => Mon
# Gold        2 Years  => 730 days  last day of the month => 01.
# Platinum    4 Years  => 1460 days last day of the quarter => 01.04. 01.07.
#
#
#DATALIST      ;DESCRIPTION                ;HOSTNAME      ;RETENTION    ;COMPRESSION   ;
DB_CEDPROD     ;Silver-Level-Backup        ;adm_oral_tst  ;150          ;NONE          ;
DB_COIPROD     ;Silver-Level-Backup        ;adm_oral_tst  ;150          ;COMPRESS      ;
DB_COIPROD     ;Gold-Level-Backup         ;adm_oral_tst  ;730          ;COMPRESS      ;
DB_COIPROD     ;Platinum-Level-Backup     ;adm_oral_tst  ;1460         ;COMPRESS      ;
DB_DMSPROD     ;Silver-Level-Backup        ;adm_oral_tst  ;150          ;BZIP2         ;
DB_DMSPROD     ;Gold-Level-Backup         ;adm_oral_tst  ;730          ;BZIP2         ;
DB_ASYPROD     ;POC-Test-Backup           ;adm_oral_tst  ;103          ;ZIP           ;
SQL_plumdb     ;Silver-Level-Backup        ;adm_oral_tst  ;150          ;ZIP           ;
SQL_CHaT       ;Silver-Level-Backup        ;adm_oral_tst  ;150          ;ZIP           ;
SQL_CHaT       ;Gold-Level-Backup         ;adm_oral_tst  ;730          ;ZIP           ;

```

SEE ALSO

edrcintro(1), **datalist.dat(4)**, **volume.dat(4)**, **lots.cfg(4)**, **lots(1m)**

NOTES

To verify the schedule definitions, use the **lots -a list_schedule** command. Only correct entries will be listed.

BUGS

-

AUTHOR

schedule.dat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

screen – screen manager with VT100/ANSI terminal emulation

SYNOPSIS

```
screen [ -options ] [ cmd [ args ] ]  
screen -r [[pid.]tty[.host]]  
screen -r sessionowner/[[pid.]tty[.host]]
```

DESCRIPTION

Screen is a full-screen window manager that multiplexes a physical terminal between several processes (typically interactive shells). Each virtual terminal provides the functions of a DEC VT100 terminal and, in addition, several control functions from the ISO 6429 (ECMA 48, ANSI X3.64) and ISO 2022 standards (e.g. insert/delete line and support for multiple character sets). There is a scrollback history buffer for each virtual terminal and a copy-and-paste mechanism that allows moving text regions between windows.

When *screen* is called, it creates a single window with a shell in it (or the specified command) and then gets out of your way so that you can use the program as you normally would. Then, at any time, you can create new (full-screen) windows with other programs in them (including more shells), kill existing windows, view a list of windows, turn output logging on and off, copy-and-paste text between windows, view the scrollback history, switch between windows in whatever manner you wish, etc. All windows run their programs completely independent of each other. Programs continue to run when their window is currently not visible and even when the whole *screen* session is detached from the user's terminal. When a program terminates, *screen* (per default) kills the window that contained it. If this window was in the foreground, the display switches to the previous window; if none are left, *screen* exits.

Everything you type is sent to the program running in the current window. The only exception to this is the one keystroke that is used to initiate a command to the window manager. By default, each command begins with a control-a (abbreviated C-a from now on), and is followed by one other keystroke. The command character and all the key bindings can be fully customized to be anything you like, though they are always two characters in length.

Screen does not understand the prefix “C-” to mean control. Please use the caret notation (“^A” instead of “C-a”) as arguments to e.g. the *escape* command or the *-e* option. *Screen* will also print out control characters in caret notation.

The standard way to create a new window is to type “C-a c”. This creates a new window running a shell and switches to that window immediately, regardless of the state of the process running in the current window. Similarly, you can create a new window with a custom command in it by first binding the command to a keystroke (in your *.screenrc* file or at the “C-a :” command line) and then using it just like the “C-a c” command. In addition, new windows can be created by running a command like:

```
screen emacs prog.c
```

from a shell prompt within a previously created window. This will not run another copy of *screen*, but will instead supply the command name and its arguments to the window manager (specified in the \$STY environment variable) who will use it to create the new window. The above example would start the emacs editor (editing prog.c) and switch to its window.

If “/etc/utmp” is writable by *screen*, an appropriate record will be written to this file for each window, and removed when the window is terminated. This is useful for working with “talk”, “script”, “shutdown”, “rsend”, “sccs” and other similar programs that use the utmp file to determine who you are. As long as *screen* is active on your terminal, the terminal's own record is removed from the utmp file. See also “C-a L”.

GETTING STARTED

Before you begin to use *screen* you'll need to make sure you have correctly selected your terminal type, just as you would for any other termcap/terminfo program. (You can do this by using *tset* for example.)

If you're impatient and want to get started without doing a lot more reading, you should remember this one command: "C-a ?". Typing these two characters will display a list of the available *screen* commands and their bindings. Each keystroke is discussed in the section "DEFAULT KEY BINDINGS". The manual section "CUSTOMIZATION" deals with the contents of your *.screenrc*.

If your terminal is a "true" auto-margin terminal (it doesn't allow the last position on the screen to be updated without scrolling the screen) consider using a version of your terminal's termcap that has automatic margins turned *off*. This will ensure an accurate and optimal update of the screen in all circumstances. Most terminals nowadays have "magic" margins (automatic margins plus usable last column). This is the VT100 style type and perfectly suited for *screen*. If all you've got is a "true" auto-margin terminal *screen* will be content to use it, but updating a character put into the last position on the screen may not be possible until the screen scrolls or the character is moved into a safe position in some other way. This delay can be shortened by using a terminal with insert-character capability.

COMMAND-LINE OPTIONS

Screen has the following command-line options:

- a** include *all* capabilities (with some minor exceptions) in each window's termcap, even if *screen* must redraw parts of the display in order to implement a function.
- A** Adapt the sizes of all windows to the size of the current terminal. By default, *screen* tries to restore its old window sizes when attaching to resizable terminals (those with "WS" in its description, e.g. *suncmd* or some *xterm*).
- c file**
override the default configuration file from "\$HOME/.screenrc" to *file*.
- d|-D [pid.tty.host]**
does not start *screen*, but detaches the elsewhere running *screen* session. It has the same effect as typing "C-a d" from *screen*'s controlling terminal. **-D** is the equivalent to the power detach key. If no session can be detached, this option is ignored. In combination with the **-r/-R** option more powerful effects can be achieved:
- d -r** Reattach a session and if necessary detach it first.
- d -R** Reattach a session and if necessary detach or even create it first.
- d -RR** Reattach a session and if necessary detach or create it. Use the first session if more than one session is available.
- D -r** Reattach a session. If necessary detach and logout remotely first.
- D -R** Attach here and now. In detail this means: If a session is running, then reattach. If necessary detach and logout remotely first. If it was not running create it and notify the user. This is the author's favorite.
- D -RR**
Attach here and now. Whatever that means, just do it.

Note: It is always a good idea to check the status of your sessions by means of "screen -list".

- e xy**
specifies the command character to be *x* and the character generating a literal command character to *y* (when typed after the command character). The default is "C-a" and 'a', which can be specified as "-e^Aa". When creating a *screen* session, this option sets the default command character. In a multi-user session all users added will start off with this command character. But when attaching to an already running session, this option changes only the command character of the attaching user. This

option is equivalent to either the commands “defescape” or “escape” respectively.

-f, -fn, and -fa

turns flow-control on, off, or “automatic switching mode”. This can also be defined through the “defflow” .screenrc command.

-h *num*

Specifies the history scrollback buffer to be *num* lines high.

-i

will cause the interrupt key (usually C-c) to interrupt the display immediately when flow-control is on. See the “defflow” .screenrc command for details. The use of this option is discouraged.

-l and -ln

turns login mode on or off (for /etc/utmp updating). This can also be defined through the “deflogin” .screenrc command.

-ls and -list

does not start *screen*, but prints a list of *pid.tty.host* strings identifying your *screen* sessions. Sessions marked ‘detached’ can be resumed with “screen -r”. Those marked ‘attached’ are running and have a controlling terminal. If the session runs in multiuser mode, it is marked ‘multi’. Sessions marked as ‘unreachable’ either live on a different host or are ‘dead’. An unreachable session is considered dead, when its name matches either the name of the local host, or the specified parameter, if any. See the **-r** flag for a description how to construct matches. Sessions marked as ‘dead’ should be thoroughly checked and removed. Ask your system administrator if you are not sure. Remove sessions with the **-wipe** option.

-L

tells *screen* to turn on automatic output logging for the windows.

-m

causes *screen* to ignore the \$STY environment variable. With “screen -m” creation of a new session is enforced, regardless whether *screen* is called from within another *screen* session or not. This flag has a special meaning in connection with the ‘-d’ option:

-d -m

Start *screen* in “detached” mode. This creates a new session but doesn’t attach to it. This is useful for system startup scripts.

-D -m

This also starts *screen* in “detached” mode, but doesn’t fork a new process. The command exits if the session terminates.

-O

selects a more optimal output mode for your terminal rather than true VT100 emulation (only affects auto-margin terminals without ‘LP’). This can also be set in your .screenrc by specifying ‘OP’ in a “termcap” command.

-p *number_or_name*

Preselect a window. This is useful when you want to reattach to a specific window or you want to send a command via the “-X” option to a specific window. As with *screen*’s select command, “-” selects the blank window. As a special case for reattach, “=” brings up the windowlist on the blank window.

-q

Suppress printing of error messages. In combination with “-ls” the exit value is as follows: 9 indicates a directory without sessions. 10 indicates a directory with running but not attachable sessions. 11 (or more) indicates 1 (or more) usable sessions. In combination with “-r” the exit value is as follows: 10 indicates that there is no session to resume. 12 (or more) indicates that there are 2 (or more) sessions to resume and you should specify which one to choose. In all other cases “-q” has no effect.

-r [*pid.tty.host*]

-r *sessionowner*/[*pid.tty.host*]

resumes a detached *screen* session. No other options (except combinations with **-d/-D**) may be specified, though an optional prefix of [*pid.tty.host*] may be needed to distinguish between multiple detached *screen* sessions. The second form is used to connect to another user’s *screen* session which runs in multiuser mode. This indicates that *screen* should look for sessions in another user’s directory. This requires setuid-root.

- R** attempts to resume the first detached *screen* session it finds. If successful, all other command-line options are ignored. If no detached session exists, starts a new session using the specified options, just as if **-R** had not been specified. The option is set by default if *screen* is run as a login-shell (actually *screen* uses “-xRR” in that case). For combinations with the **-d/-D** option see there.
- s** sets the default shell to the program specified, instead of the value in the environment variable \$SHELL (or “/bin/sh” if not defined). This can also be defined through the “shell” *.screenrc* command.
- S sessionname**
When creating a new session, this option can be used to specify a meaningful name for the session. This name identifies the session for “screen -list” and “screen -r” actions. It substitutes the default [tty.host] suffix.
- t name**
sets the title (a.k.a.) for the default shell or specified program. See also the “shelltitle” *.screenrc* command.
- U** Run *screen* in UTF-8 mode. This option tells *screen* that your terminal sends and understands UTF-8 encoded characters. It also sets the default encoding for new windows to ‘utf8’.
- v** Print version number.
- wipe [match]**
does the same as “screen -ls”, but removes destroyed sessions instead of marking them as ‘dead’. An unreachable session is considered dead, when its name matches either the name of the local host, or the explicitly given parameter, if any. See the **-r** flag for a description how to construct matches.
- x** Attach to a not detached *screen* session. (Multi display mode).
- X** Send the specified command to a running *screen* session. You can use the **-d** or **-r** option to tell *screen* to look only for attached or detached *screen* sessions. Note that this command doesn’t work if the session is password protected.

DEFAULT KEY BINDINGS

As mentioned, each *screen* command consists of a “C-a” followed by one other character. For your convenience, all commands that are bound to lower-case letters are also bound to their control character counterparts (with the exception of “C-a a”; see below), thus, “C-a c” as well as “C-a C-c” can be used to create a window. See section “CUSTOMIZATION” for a description of the command.

The following table shows the default key bindings:

| | | |
|----------------|-----------------|--|
| C-a ’ | (select) | Prompt for a window name or number to switch to. |
| C-a " | (windowlist -b) | Present a list of all windows for selection. |
| C-a 0 | (select 0) | |
| ... | ... | |
| C-a 9 | (select 9) | |
| C-a - | (select -) | Switch to window number 0 – 9, or to the blank window. |
| C-a tab | (focus) | Switch the input focus to the next region. |
| C-a C-a | (other) | Toggle to the window displayed previously. Note that this binding defaults to the command character typed twice, unless overridden. For instance, if you use the option “-e x”, this command becomes “]]”. |
| C-a a | (meta) | Send the command character (C-a) to window. See <i>escape</i> command. |
| C-a A | (title) | Allow the user to enter a name for the current window. |
| C-a b | | |

| | | |
|----------------------|--------------|--|
| C-a C-b | (break) | Send a break to window. |
| C-a B | (pow_break) | Reopen the terminal line and send a break. |
| C-a c | | |
| C-a C-c | (screen) | Create a new window with a shell and switch to that window. |
| C-a C | (clear) | Clear the screen. |
| C-a d | | |
| C-a C-d | (detach) | Detach <i>screen</i> from this terminal. |
| C-a D D | (pow_detach) | Detach and logout. |
| C-a f | | |
| C-a C-f | (flow) | Toggle flow <i>on</i> , <i>off</i> or <i>auto</i> . |
| C-a F | (fit) | Resize the window to the current region size. |
| C-a C-g | (vbell) | Toggles <i>screen</i> 's visual bell mode. |
| C-a h | (hardcopy) | Write a hardcopy of the current window to the file “hardcopy. <i>n</i> ”. |
| C-a H | (log) | Begins/ends logging of the current window to the file “screenlog. <i>n</i> ”. |
| C-a i | | |
| C-a C-i | (info) | Show info about this window. |
| C-a k | | |
| C-a C-k | (kill) | Destroy current window. |
| C-a l | | |
| C-a C-l | (redisplay) | Fully refresh current window. |
| C-a L | (login) | Toggle this windows login slot. Available only if <i>screen</i> is configured to update the utmp database. |
| C-a m | | |
| C-a C-m | (lastmsg) | Repeat the last message displayed in the message line. |
| C-a M | (monitor) | Toggles monitoring of the current window. |
| C-a space | | |
| C-a n | | |
| C-a C-n | (next) | Switch to the next window. |
| C-a N | (number) | Show the number (and title) of the current window. |
| C-a backspace | | |
| C-a h | | |
| C-a p | | |
| C-a C-p | (prev) | Switch to the previous window (opposite of C-a n). |
| C-a q | | |
| C-a C-q | (xon) | Send a control-q to the current window. |
| C-a Q | (only) | Delete all regions but the current one. |
| C-a r | | |
| C-a C-r | (wrap) | Toggle the current window's line-wrap setting (turn the current window's automatic margins on and off). |
| C-a s | | |
| C-a C-s | (xoff) | Send a control-s to the current window. |
| C-a S | (split) | Split the current region into two new ones. |
| C-a t | | |

| | | |
|-----------------|---------------|--|
| C-a C-t | (time) | Show system information. |
| C-a v | (version) | Display the version and compilation date. |
| C-a C-v | (digraph) | Enter digraph. |
| C-a w | | |
| C-a C-w | (windows) | Show a list of window. |
| C-a W | (width) | Toggle 80/132 columns. |
| C-a x | | |
| C-a C-x | (lockscreen) | Lock this terminal. |
| C-a X | (remove) | Kill the current region. |
| C-a z | | |
| C-a C-z | (suspend) | Suspend <i>screen</i> . Your system must support BSD-style job-control. |
| C-a Z | (reset) | Reset the virtual terminal to its “power-on” values. |
| C-a . | (dumftermcap) | Write out a “.termcap” file. |
| C-a ? | (help) | Show key bindings. |
| C-a C-\ | (quit) | Kill all windows and terminate <i>screen</i> . |
| C-a : | (colon) | Enter command line mode. |
| C-a [| | |
| C-a C-[| | |
| C-a esc | (copy) | Enter copy/scrollback mode. |
| C-a] | (paste .) | Write the contents of the paste buffer to the stdin queue of the current window. |
| C-a { | | |
| C-a } | (history) | Copy and paste a previous (command) line. |
| C-a > | (writebuf) | Write paste buffer to a file. |
| C-a < | (readbuf) | Reads the screen-exchange file into the paste buffer. |
| C-a = | (removebuf) | Removes the file used by C-a < and C-a > . |
| C-a , | (license) | Shows where <i>screen</i> comes from, where it went to and why you can use it. |
| C-a _ | (silence) | Start/stop monitoring the current window for inactivity. |
| C-a * | (displays) | Show a listing of all currently attached displays. |

CUSTOMIZATION

The “socket directory” defaults either to \$HOME/.screen or simply to /tmp/screens or preferably to /usr/local/screens chosen at compile-time. If *screen* is installed setuid-root, then the administrator should compile *screen* with an adequate (not NFS mounted) socket directory. If *screen* is not running setuid-root, the user can specify any mode 700 directory in the environment variable \$SCREENDIR.

When *screen* is invoked, it executes initialization commands from the files “/etc/screenrc” and “.screenrc” in the user’s home directory. These are the “programmer’s defaults” that can be overridden in the following ways: for the global screenrc file *screen* searches for the environment variable \$SYSSCREENRC (this override feature may be disabled at compile-time). The user specific screenrc file is searched in \$SCREENRC, then \$HOME/.screenrc. The command line option **-c** takes precedence over the above user screenrc files.

Commands in these files are used to set options, bind functions to keys, and to automatically establish one or more windows at the beginning of your *screen* session. Commands are listed one per line, with empty lines being ignored. A command’s arguments are separated by tabs or spaces, and may be surrounded by

single or double quotes. A '#' turns the rest of the line into a comment, except in quotes. Unintelligible lines are warned about and ignored. Commands may contain references to environment variables. The syntax is the shell-like "\$VAR " or "\${VAR} ". Note that this causes incompatibility with previous *screen* versions, as now the '\$'-character has to be protected with '\' if no variable substitution shall be performed. A string in single-quotes is also protected from variable substitution.

Two configuration files are shipped as examples with your *screen* distribution: "etc/screenrc" and "etc/etc-screenrc". They contain a number of useful examples for various commands.

Customization can also be done 'on-line'. To enter the command mode type 'C-a :'. Note that commands starting with "def" change default values, while others change current settings.

The following commands are available:

acladd *usernames* [*crypted-pw*]

addacl *usernames*

Enable users to fully access this *screen* session. *Usernames* can be one user or a comma separated list of users. This command enables to attach to the *screen* session and performs the equivalent of 'aclchg *usernames* +rwx "#?"'. executed. To add a user with restricted access, use the 'aclchg' command below. If an optional second parameter is supplied, it should be a crypted password for the named user(s). 'Addacl' is a synonym to 'acladd'. Multi user mode only.

aclchg *usernames* *permbits* *list*

chacl *usernames* *permbits* *list*

Change permissions for a comma separated list of users. Permission bits are represented as 'r', 'w' and 'x'. Prefixing '+' grants the permission, '-' removes it. The third parameter is a comma separated list of commands and/or windows (specified either by number or title). The special list '#' refers to all windows, '?' to all commands. if *usernames* consists of a single '*', all known users are affected. A command can be executed when the user has the 'x' bit for it. The user can type input to a window when he has its 'w' bit set and no other user obtains a writelock for this window. Other bits are currently ignored. To withdraw the writelock from another user in window 2: 'aclchg *username* -w+w 2'. To allow read-only access to the session: 'aclchg *username* -w "#?". As soon as a user's name is known to *screen* he can attach to the session and (per default) has full permissions for all command and windows. Execution permission for the acl commands, 'at' and others should also be removed or the user may be able to regain write permission. Rights of the special username **nobody** cannot be changed (see the "su" command). 'Chacl' is a synonym to 'aclchg'. Multi user mode only.

acldel *username*

Remove a user from *screen*'s access control list. If currently attached, all the user's displays are detached from the session. He cannot attach again. Multi user mode only.

aclgrp *username* [*groupname*]

Creates groups of users that share common access rights. The name of the group is the username of the group leader. Each member of the group inherits the permissions that are granted to the group leader. That means, if a user fails an access check, another check is made for the group leader. A user is removed from all groups the special value "none" is used for *groupname*. If the second parameter is omitted all groups the user is in are listed.

aclumask [[*users*]+*bits* [[*users*]-*bits*]

umask [[*users*]+*bits* [[*users*]-*bits*]

This specifies the access other users have to windows that will be created by the caller of the command. *Users* may be no, one or a comma separated list of known usernames. If no users are specified, a list of all currently known users is assumed. *Bits* is any combination of access control bits allowed defined with the "aclchg" command. The special username "?" predefines the access that not yet known users will be

granted to any window initially. The special username “??” predefines the access that not yet known users are granted to any command. Rights of the special username **nobody** cannot be changed (see the “su” command). ‘Umask’ is a synonym to ‘aclumask’.

activity *message*

When any activity occurs in a background window that is being monitored, *screen* displays a notification in the message line. The notification message can be re-defined by means of the “activity” command. Each occurrence of ‘%’ in *message* is replaced by the number of the window in which activity has occurred, and each occurrence of ‘^G’ is replaced by the definition for bell in your termcap (usually an audible bell). The default message is

’Activity in window %n’

Note that monitoring is off for all windows by default, but can be altered by use of the “monitor” command (C-a M).

allpartial on|off

If set to on, only the current cursor line is refreshed on window change. This affects all windows and is useful for slow terminal lines. The previous setting of full/partial refresh for each window is restored with “allpartial off”. This is a global flag that immediately takes effect on all windows overriding the “partial” settings. It does not change the default redraw behavior of newly created windows.

altscreen on|off

If set to on, “alternate screen” support is enabled in virtual terminals, just like in xterm. Initial setting is ‘off’.

at [*identifier*][#|*|%] *command* [*args* ...]

Execute a command at other displays or windows as if it had been entered there. “At” changes the context (the ‘current window’ or ‘current display’ setting) of the command. If the first parameter describes a non-unique context, the command will be executed multiple times. If the first parameter is of the form ‘*identifier**’ then *identifier* is matched against user names. The command is executed once for each display of the selected user(s). If the first parameter is of the form ‘*identifier*%’ *identifier* is matched against displays. Displays are named after the ttys they attach. The prefix ‘/dev/’ or ‘/dev/tty’ may be omitted from the identifier. If *identifier* has a ‘#’ or nothing appended it is matched against window numbers and titles. Omitting an identifier in front of the ‘#’, ‘*’ or ‘%’-character selects all users, displays or windows because a prefix-match is performed. Note that on the affected display(s) a short message will describe what happened. Permission is checked for initiator of the “at” command, not for the owners of the affected display(s). Note that the ‘#’ character works as a comment introducer when it is preceded by whitespace. This can be escaped by prefixing a ‘\’. Permission is checked for the initiator of the “at” command, not for the owners of the affected display(s).

Caveat: When matching against windows, the command is executed at least once per window. Commands that change the internal arrangement of windows (like “other”) may be called again. In shared windows the command will be repeated for each attached display. Beware, when issuing toggle commands like “login”! Some commands (e.g. “process”) require that a display is associated with the target windows. These commands may not work correctly under “at” looping over windows.

attrcolor *attrib* [*attribute/color-modifier*]

This command can be used to highlight attributes by changing the color of the text. If the attribute *attrib* is in use, the specified attribute/color modifier is also applied. If no modifier is given, the current one is deleted. See the “STRING ESCAPES” chapter for the syntax of the modifier. Screen understands two pseudo-attributes, “i” stands for high-intensity foreground color and “I” for high-intensity background color.

Examples:

```
attrcolor b "R"
```

Change the color to bright red if bold text is to be printed.

```
attrcolor u "-u b"
```

Use blue text instead of underline.

```
attrcolor b ".I"
```

Use bright colors for bold text. Most terminal emulators do this already.

```
attrcolor i "+b"
```

Make bright colored text also bold.

autodetach on|off

Sets whether *screen* will automatically detach upon hangup, which saves all your running programs until they are resumed with a **screen -r** command. When turned off, a hangup signal will terminate *screen* and all the processes it contains. Autodetach is on by default.

autonuke on|off

Sets whether a clear screen sequence should nuke all the output that has not been written to the terminal. See also “obuflimit”.

backtick *id lifespan autorefresh cmd args...*

backtick *id*

Program the backtick command with the numerical id *id*. The output of such a command is used for substitution of the “%” string escape. The specified *lifespan* is the number of seconds the output is considered valid. After this time, the command is run again if a corresponding string escape is encountered. The *autorefresh* parameter triggers an automatic refresh for caption and hardstatus strings after the specified number of seconds. Only the last line of output is used for substitution.

If both the *lifespan* and the *autorefresh* parameters are zero, the backtick program is expected to stay in the background and generate output once in a while. In this case, the command is executed right away and *screen* stores the last line of output. If a new line gets printed *screen* will automatically refresh the hardstatus or the captions.

The second form of the command deletes the backtick command with the numerical id *id*.

bce [on|off]

Change background-color-erase setting. If “bce” is set to on, all characters cleared by an erase/insert/scroll/clear operation will be displayed in the current background color. Otherwise the default background color is used.

bell_msg [message]

When a bell character is sent to a background window, *screen* displays a notification in the message line. The notification message can be re-defined by this command. Each occurrence of ‘%’ in *message* is replaced by the number of the window to which a bell has been sent, and each occurrence of ‘^G’ is replaced by the definition for bell in your termcap (usually an audible bell). The default message is

```
'Bell in window %n'
```

An empty message can be supplied to the “bell_msg” command to suppress output of a message line (bell_msg ""). Without parameter, the current message is shown.

bind [-c *class*] *key* [*command* [*args*]]

Bind a command to a key. By default, most of the commands provided by *screen* are bound to one or more keys as indicated in the “DEFAULT KEY BINDINGS” section, e. g. the command to create a new window is bound to “C-c” and “c”. The “bind” command can be used to redefine the key bindings and to define new bindings. The *key* argument is either a single character, a two-character sequence of the form “^x” (meaning “C-x”), a backslash followed by an octal number (specifying the ASCII code of the character), or a backslash followed by a second character, such as “\^” or “\\”. The argument can also be quoted, if you like. If no further argument is given, any previously established binding for this key is removed. The *command* argument can be any command listed in this section.

If a command class is specified via the “-c” option, the key is bound for the specified class. Use the “command” command to activate a class. Command classes can be used to create multiple command keys or multi-character bindings.

Some examples:

```
bind ' ' windows
bind ^k
bind k
bind K kill
bind ^f screen telnet foobar
bind \033 screen -ln -t root -h 1000 9 su
```

would bind the space key to the command that displays a list of windows (so that the command usually invoked by “C-a C-w” would also be available as “C-a space”). The next three lines remove the default kill binding from “C-a C-k” and “C-a k”. “C-a K” is then bound to the kill command. Then it binds “C-f” to the command “create a window with a TELNET connection to foobar”, and bind “escape” to the command that creates an non-login window with a.k.a. “root” in slot #9, with a superuser shell and a scrollbar buffer of 1000 lines.

```
bind -c demo1 0 select 10
bind -c demo1 1 select 11
bind -c demo1 2 select 12
bindkey ""B" command -c demo1
```

makes “C-b 0” select window 10, “C-b 1” window 11, etc.

```
bind -c demo2 0 select 10
bind -c demo2 1 select 11
bind -c demo2 2 select 12
bind - command -c demo2
```

makes “C-a - 0” select window 10, “C-a - 1” window 11, etc.

bindkey [-d] [-m] [-a] [[-k|-t] *string* [*cmd* *args*]]

This command manages screen’s input translation tables. Every entry in one of the tables tells screen how to react if a certain sequence of characters is encountered. There are three tables: one that should contain actions programmed by the user, one for the default actions used for terminal emulation and one for screen’s copy mode to do cursor movement. See section “INPUT TRANSLATION” for a list of default key bindings.

If the **-d** option is given, bindkey modifies the default table, **-m** changes the copy mode table and with neither option the user table is selected. The argument *string* is the sequence of characters to which an action is bound. This can either be a fixed string or a termcap keyboard capability name (selectable with the **-k** option).

Some keys on a VT100 terminal can send a different string if application mode is turned on (e.g the cursor keys). Such keys have two entries in the translation table. You can select the application mode entry by specifying the **-a** option.

The **-t** option tells screen not to do inter-character timing. One cannot turn off the timing if a termcap

capability is used.

Cmd can be any of screen's commands with an arbitrary number of *args*. If *cmd* is omitted the key-binding is removed from the table.

Here are some examples of keyboard bindings:

```
bindkey -d
```

Show all of the default key bindings. The application mode entries are marked with [A].

```
bindkey -k k1 select 1
```

Make the "F1" key switch to window one.

```
bindkey -t foo stuff barfoo
```

Make "foo" an abbreviation of the word "barfoo". Timeout is disabled so that users can type slowly.

```
bindkey "\024" mapdefault
```

This key-binding makes “^T” an escape character for key-bindings. If you did the above “stuff barfoo” binding, you can enter the word “foo” by typing “^Tfoo”. If you want to insert a “^T” you have to press the key twice (i.e. escape the escape binding).

```
bindkey -k F1 command
```

Make the F11 (not F1!) key an alternative screen escape (besides ^A).

break [*duration*]

Send a break signal for *duration**0.25 seconds to this window. For non-Posix systems the time interval may be rounded up to full seconds. Most useful if a character device is attached to the window rather than a shell process (See also chapter “WINDOW TYPES”). The maximum duration of a break signal is limited to 15 seconds.

blanker

Activate the screen blanker. First the screen is cleared. If no blanker program is defined, the cursor is turned off, otherwise, the program is started and its output is written to the screen. The screen blanker is killed with the first keypress, the read key is discarded.

This command is normally used together with the “idle” command.

blankerprg [*program args*]

Defines a blanker program. Disables the blanker program if no arguments are given.

breaktype [*tcsendbreak*|*TIOCSBRK*|*TCSBRK*]

Choose one of the available methods of generating a break signal for terminal devices. This command should affect the current window only. But it still behaves identical to “defbreaktype”. This will be changed in the future. Calling “breaktype” with no parameter displays the break method for the current window.

bufferfile [*exchange-file*]

Change the filename used for reading and writing with the paste buffer. If the optional argument to the “bufferfile” command is omitted, the default setting (“/tmp/screen-exchange”) is reactivated. The following example will paste the system's password file into the *screen* window (using the paste buffer, where a copy remains):

```
C-a : bufferfile /etc/passwd
C-a < C-a ]
C-a : bufferfile
```

c1 [**on|off**]

Change c1 code processing. “C1 on” tells screen to treat the input characters between 128 and 159 as control functions. Such an 8-bit code is normally the same as ESC followed by the corresponding 7-bit code. The default setting is to process c1 codes and can be changed with the “defc1” command. Users with fonts that have usable characters in the c1 positions may want to turn this off.

caption always|splitonly [*string*]**caption string** [*string*]

This command controls the display of the window captions. Normally a caption is only used if more than one window is shown on the display (split screen mode). But if the type is set to **always** screen shows a caption even if only one window is displayed. The default is **splitonly**.

The second form changes the text used for the caption. You can use all escapes from the “STRING ESCAPES” chapter. Screen uses a default of ‘%3n %t’.

You can mix both forms by providing a string as an additional argument.

charset *set*

Change the current character set slot designation and charset mapping. The first four character of *set* are treated as charset designators while the fifth and sixth character must be in range ‘0’ to ‘3’ and set the GL/GR charset mapping. On every position a ‘.’ may be used to indicate that the corresponding charset/mapping should not be changed (*set* is padded to six characters internally by appending ‘.’ chars). New windows have “BBBB02” as default charset, unless a “encoding” command is active.

The current setting can be viewed with the “info” command.

chdir [*directory*]

Change the *current directory* of *screen* to the specified directory or, if called without an argument, to your home directory (the value of the environment variable \$HOME). All windows that are created by means of the “screen” command from within “.screenrc” or by means of “C-a : screen ...” or “C-a c” use this as their default directory. Without a chdir command, this would be the directory from which *screen* was invoked. Hardcopy and log files are always written to the *window*’s default directory, *not* the current directory of the process running in the window. You can use this command multiple times in your .screenrc to start various windows in different default directories, but the last chdir value will affect all the windows you create interactively.

clear

Clears the current window and saves its image to the scrollbar buffer.

colon [*prefix*]

Allows you to enter “.screenrc” command lines. Useful for on-the-fly modification of key bindings, specific window creation and changing settings. Note that the “set” keyword no longer exists! Usually commands affect the current window rather than default settings for future windows. Change defaults with commands starting with ‘def...’.

If you consider this as the ‘Ex command mode’ of *screen*, you may regard “C-a esc” (copy mode) as its ‘Vi command mode’.

command [**-c** *class*]

This command has the same effect as typing the screen escape character (^A). It is probably only useful for key bindings. If the “-c” option is given, select the specified command class. See also “bind” and “bind-key”.

compacthist [on|off]

This tells screen whether to suppress trailing blank lines when scrolling up text into the history buffer.

console [on|off]

Grabs or un-grabs the machines console output to a window. *Note:* Only the owner of `/dev/console` can grab the console output. This command is only available if the machine supports the ioctl `TIOCCONS`.

copy

Enter copy/scrollback mode. This allows you to copy text from the current window and its history into the paste buffer. In this mode a vi-like ‘full screen editor’ is active:

Movement keys:

- h, j, k, l** move the cursor line by line or column by column.
- 0, ^** and **\$** move to the leftmost column, to the first or last non-whitespace character on the line.
- H, M** and **L** move the cursor to the leftmost column of the top, center or bottom line of the window.
- +** and **-** positions one line up and down.
- G** moves to the specified absolute line (default: end of buffer).
- |** moves to the specified absolute column.
- w, b, e** move the cursor word by word.
- B, E** move the cursor WORD by WORD (as in vi).
- C-u** and **C-d** scroll the display up/down by the specified amount of lines while preserving the cursor position. (Default: half screen-full).
- C-b** and **C-f** scroll the display up/down a full screen.
- g** moves to the beginning of the buffer.
- %** jumps to the specified percentage of the buffer.

Note:

Emacs style movement keys can be customized by a `.screenrc` command. (E.g. `markkeys "h=^B:l=^F:$=^E"`) There is no simple method for a full emacs-style keymap, as this involves multi-character codes.

Marking:

The copy range is specified by setting two marks. The text between these marks will be highlighted. Press

- space** to set the first or second mark respectively.
- Y** and **y** used to mark one whole line or to mark from start of line.
- W** marks exactly one word.

Repeat count:

Any of these commands can be prefixed with a repeat count number by pressing digits

- 0..9** which is taken as a repeat count.
- Example: “`C-a C-[H 10 j 5 Y`” will copy lines 11 to 15 into the paste buffer.

Searching:

- /** Vi-like search forward.
- ?** Vi-like search backward.
- C-a s** Emacs style incremental search forward.
- C-r** Emacs style reverse i-search.

Specials:

There are however some keys that act differently than in *vi*. *Vi* does not allow one to yank rectangular blocks of text, but *screen* does. Press

- c** or **C** to set the left or right margin respectively. If no repeat count is given, both default to the current cursor position.

Example: Try this on a rather full text screen: “`C-a [M 20 l SPACE c 10 l 5 j C SPACE`”.

This moves one to the middle line of the screen, moves in 20 columns left, marks the beginning of the

paste buffer, sets the left column, moves 5 columns down, sets the right column, and then marks the end of the paste buffer. Now try:

```
“C-a [ M 20 l SPACE 10 l 5 j SPACE”
```

and notice the difference in the amount of text copied.

J joins lines. It toggles between 4 modes: lines separated by a newline character (012), lines glued seamless, lines separated by a single whitespace and comma separated lines. Note that you can prepend the newline character with a carriage return character, by issuing a “crlf on”.

v is for all the *vi* users with “:set numbers” – it toggles the left margin between column 9 and 1. Press **a** before the final space key to toggle in append mode. Thus the contents of the paste buffer will not be overwritten, but is appended to.

A toggles in append mode and sets a (second) mark.

> sets the (second) mark and writes the contents of the paste buffer to the screen-exchange file (/tmp/screen-exchange per default) once copy-mode is finished.

This example demonstrates how to dump the whole scrollback buffer to that file: “C-A [g SPACE G \$ >”.

C-g gives information about the current line and column.

x exchanges the first mark and the current cursor position. You can use this to adjust an already placed mark.

@ does nothing. Does not even exit copy mode.

All keys not described here exit copy mode.

copy_reg [*key*]

No longer exists, use “readreg” instead.

crlf [on|off]

This affects the copying of text regions with the ‘C-a [’ command. If it is set to ‘on’, lines will be separated by the two character sequence ‘CR’ - ‘LF’. Otherwise (default) only ‘LF’ is used. When no parameter is given, the state is toggled.

debug on|off

Turns runtime debugging on or off. If *screen* has been compiled with option -DDEBUG debugging available and is turned on per default. Note that this command only affects debugging output from the main “SCREEN” process correctly. Debug output from attacher processes can only be turned off once and forever.

defc1 on|off

Same as the **c1** command except that the default setting for new windows is changed. Initial setting is ‘on’.

defautonuke on|off

Same as the **autonuke** command except that the default setting for new displays is changed. Initial setting is ‘off’. Note that you can use the special ‘AN’ terminal capability if you want to have a dependency on the terminal type.

defbce on|off

Same as the **bce** command except that the default setting for new windows is changed. Initial setting is ‘off’.

defbreaktype [*tcsendbreak*|*TIOCSBRK*|*TCSBRK*]

Choose one of the available methods of generating a break signal for terminal devices. The preferred methods are *tcsendbreak* and *TIOCSBRK*. The third, *TCSBRK*, blocks the complete *screen* session for the duration of the break, but it may be the only way to generate long breaks. *Tcsendbreak* and *TIOCSBRK* may or

may not produce long breaks with spikes (e.g. 4 per second). This is not only system dependant, this also differs between serial board drivers. Calling “defbreaktype” with no parameter displays the current setting.

defcharset [*set*]

Like the **charset** command except that the default setting for new windows is changed. Shows current default if called without argument.

defescape *xy*

Set the default command characters. This is equivalent to the “escape” except that it is useful multiuser sessions only. In a multiuser session “escape” changes the command character of the calling user, where “defescape” changes the default command characters for users that will be added later.

defflow on|off|auto [**interrupt**]

Same as the **flow** command except that the default setting for new windows is changed. Initial setting is ‘auto’. Specifying “defflow auto interrupt” is the same as the command-line options **-fa** and **-i**.

defgr on|off

Same as the **gr** command except that the default setting for new windows is changed. Initial setting is ‘off’.

defhstatus [*status*]

The hardstatus line that all new windows will get is set to *status*. This command is useful to make the hardstatus of every window display the window number or title or the like. *Status* may contain the same directives as in the window messages, but the directive escape character is ‘^E’ (octal 005) instead of ‘%’. This was done to make a misinterpretation of program generated hardstatus lines impossible. If the parameter *status* is omitted, the current default string is displayed. Per default the hardstatus line of new windows is empty.

defencoding *enc*

Same as the **encoding** command except that the default setting for new windows is changed. Initial setting is the encoding taken from the terminal.

deflog on|off

Same as the **log** command except that the default setting for new windows is changed. Initial setting is ‘off’.

deflogin on|off

Same as the **login** command except that the default setting for new windows is changed. This is initialized with ‘on’ as distributed (see config.h.in).

defmode *mode*

The mode of each newly allocated pseudo-tty is set to *mode*. *Mode* is an octal number. When no “defmode” command is given, mode 0622 is used.

defmonitor on|off

Same as the **monitor** command except that the default setting for new windows is changed. Initial setting is ‘off’.

defnonblock on|off*numsecs*

Same as the **nonblock** command except that the default setting for displays is changed. Initial setting is 'off'.

defobuflimit *limit*

Same as the **obuflimit** command except that the default setting for new displays is changed. Initial setting is 256 bytes. Note that you can use the special 'OL' terminal capability if you want to have a dependency on the terminal type.

defscrollback *num*

Same as the **scrollback** command except that the default setting for new windows is changed. Initial setting is 100.

defshell *command*

Synonym to the **shell** command. See there.

defsilence on|off

Same as the **silence** command except that the default setting for new windows is changed. Initial setting is 'off'.

defslowpaste *msec*

Same as the **slowpaste** command except that the default setting for new windows is changed. Initial setting is 0 milliseconds, meaning 'off'.

defutf8 on|off

Same as the **utf8** command except that the default setting for new windows is changed. Initial setting is 'on' if screen was started with "-U", otherwise 'off'.

defwrap on|off

Same as the **wrap** command except that the default setting for new windows is changed. Initially line-wrap is on and can be toggled with the "wrap" command ("C-a r") or by means of "C-a : wrap on|off".

defwritelock on|off|auto

Same as the **writelock** command except that the default setting for new windows is changed. Initially write-locks will off.

defzombie [*keys*]

Synonym to the **zombie** command. Both currently change the default. See there.

detach [-h]

Detach the *screen* session (disconnect it from the terminal and put it into the background). This returns you to the shell where you invoked *screen*. A detached *screen* can be resumed by invoking *screen* with the **-r** option (see also section "COMMAND-LINE OPTIONS"). The **-h** option tells screen to immediately close the connection to the terminal ("hangup").

dinfo

Show what screen thinks about your terminal. Useful if you want to know why features like color or the alternate charset don't work.

displays

Shows a tabular listing of all currently connected user front-ends (displays). This is most useful for multi-user sessions.

digraph [*preset*]

This command prompts the user for a digraph sequence. The next two characters typed are looked up in a builtin table and the resulting character is inserted in the input stream. For example, if the user enters 'a', an a-umlaut will be inserted. If the first character entered is a 0 (zero), *screen* will treat the following characters (up to three) as an octal number instead. The optional argument *preset* is treated as user input, thus one can create an "umlaut" key. For example the command "bindkey ^K digraph ''" enables the user to generate an a-umlaut by typing CTRL-K a.

dumftermcap

Write the termcap entry for the virtual terminal optimized for the currently active window to the file ".termcap" in the user's "\$HOME/.screen" directory (or wherever *screen* stores its sockets. See the "FILES" section below). This termcap entry is identical to the value of the environment variable \$TERMCAP that is set up by *screen* for each window. For terminfo based systems you will need to run a converter like *cap-toinfo* and then compile the entry with *tic*.

echo [-n] *message*

The echo command may be used to annoy *screen* users with a 'message of the day'. Typically installed in a global /etc/screenrc. The option "-n" may be used to suppress the line feed. See also "sleep". Echo is also useful for online checking of environment variables.

encoding *enc* [*enc*]

Tell *screen* how to interpret the input/output. The first argument sets the encoding of the current window. Each window can emulate a different encoding. The optional second parameter overwrites the encoding of the connected terminal. It should never be needed as *screen* uses the locale setting to detect the encoding. There is also a way to select a terminal encoding depending on the terminal type by using the "KJ" termcap entry.

Supported encodings are eucJP, SJIS, eucKR, eucCN, Big5, GBK, KOI8-R, CP1251, UTF-8, ISO8859-2, ISO8859-3, ISO8859-4, ISO8859-5, ISO8859-6, ISO8859-7, ISO8859-8, ISO8859-9, ISO8859-10, ISO8859-15, jis.

See also "defencoding", which changes the default setting of a new window.

escape *xy*

Set the command character to *x* and the character generating a literal command character (by triggering the "meta" command) to *y* (similar to the -e option). Each argument is either a single character, a two-character sequence of the form "^x" (meaning "C-x"), a backslash followed by an octal number (specifying the ASCII code of the character), or a backslash followed by a second character, such as "\^" or "\\\". The default is "^Aa".

eval *command1* [*command2* ...]

Parses and executes each argument as separate command.

exec [[*fdpat*] *newcommand* [*args* ...]]

Run a unix subprocess (specified by an executable path *newcommand* and its optional arguments) in the current window. The flow of data between newcommands stdin/stdout/stderr, the process originally started in the window (let us call it "application-process") and *screen* itself (window) is controlled by the

filedescriptor pattern *fdpat*. This pattern is basically a three character sequence representing *stdin*, *stdout* and *stderr* of *newcommand*. A dot (.) connects the file descriptor to *screen*. An exclamation mark (!) causes the file descriptor to be connected to the application-process. A colon (:) combines both. User input will go to *newcommand* unless *newcommand* receives the application-process' output (*fdpats* first character is '!' or ':' or a pipe symbol (|) is added (as a fourth character) to the end of *fdpat*.

Invoking 'exec' without arguments shows name and arguments of the currently running subprocess in this window. Only one subprocess a time can be running in each window.

When a subprocess is running the 'kill' command will affect it instead of the windows process.

Refer to the postscript file 'doc/fdpat.ps' for a confusing illustration of all 21 possible combinations. Each drawing shows the digits 2,1,0 representing the three file descriptors of *newcommand*. The box marked 'W' is the usual *pty* that has the application-process on its slave side. The box marked 'P' is the secondary *pty* that now has *screen* at its master side.

Abbreviations:

Whitespace between the word 'exec' and *fdpat* and the command can be omitted. Trailing dots and a *fdpat* consisting only of dots can be omitted. A simple '|' is synonymous for the pattern '!.|'; the word *exec* can be omitted here and can always be replaced by '!'.

Examples:

```
exec ... /bin/sh
exec /bin/sh
!/bin/sh
```

Creates another shell in the same window, while the original shell is still running. Output of both shells is displayed and user input is sent to the new */bin/sh*.

```
exec !.. stty 19200
exec ! stty 19200
!!stty 19200
```

Set the speed of the window's *tty*. If your *stty* command operates on *stdout*, then add another '!'.

```
exec !..| less
|less
```

This adds a pager to the window output. The special character '|' is needed to give the user control over the pager although it gets its input from the window's process. This works, because *less* listens on *stderr* (a behavior that *screen* would not expect without the '|') when its *stdin* is not a *tty*. *Less* versions newer than 177 fail miserably here; good old *pg* still works.

```
!:sed -n s/. *Error.*\007/p
```

Sends window output to both, the user and the *sed* command. The *sed* inserts an additional bell character (oct. 007) to the window output seen by *screen*. This will cause "Bell in window x" messages, whenever the string "Error" appears in the window.

fit

Change the window size to the size of the current region. This command is needed because *screen* doesn't adapt the window size automatically if the window is displayed more than once.

flow [on|off|auto]

Sets the flow-control mode for this window. Without parameters it cycles the current window's flow-control setting from "automatic" to "on" to "off". See the discussion on "FLOW-CONTROL" later on in this document for full details and note, that this is subject to change in future releases. Default is set by 'def-flow'.

focus [**up**|**down**|**top**|**bottom**]

Move the input focus to the next region. This is done in a cyclic way so that the top region is selected after the bottom one. If no subcommand is given it defaults to 'down'. 'up' cycles in the opposite order, 'top' and 'bottom' go to the top and bottom region respectively. Useful bindings are (j and k as in vi)

```
bind j focus down
bind k focus up
bind t focus top
bind b focus bottom
```

gr [**on**|**off**]

Turn GR charset switching on/off. Whenever screen sees an input character with the 8th bit set, it will use the charset stored in the GR slot and print the character with the 8th bit stripped. The default (see also "defgr") is not to process GR switching because otherwise the ISO88591 charset would not work.

hardcopy [**-h**] [*file*]

Writes out the currently displayed image to the file *file*, or, if no filename is specified, to *hardcopy.n* in the default directory, where *n* is the number of the current window. This either appends or overwrites the file if it exists. See below. If the option **-h** is specified, dump also the contents of the scrollbar buffer.

hardcopy_append **on**|**off**

If set to "on", *screen* will append to the "hardcopy.n" files created by the command "C-a h", otherwise these files are overwritten each time. Default is 'off'.

hardcopydir *directory*

Defines a directory where hardcopy files will be placed. If unset, hardcopies are dumped in *screen*'s current working directory.

hardstatus [**on**|**off**]**hardstatus** [**always**]*lastline*|*message*|*ignore* [*string*]**hardstatus string** [*string*]

This command configures the use and emulation of the terminal's hardstatus line. The first form toggles whether *screen* will use the hardware status line to display messages. If the flag is set to 'off', these messages are overlaid in reverse video mode at the display line. The default setting is 'on'.

The second form tells *screen* what to do if the terminal doesn't have a hardstatus line (i.e. the termcap/terminfo capabilities "hs", "ts", "fs" and "ds" are not set). If the type "lastline" is used, *screen* will reserve the last line of the display for the hardstatus. "message" uses *screen*'s message mechanism and "ignore" tells *screen* never to display the hardstatus. If you prepend the word "always" to the type (e.g., "alwayslastline"), *screen* will use the type even if the terminal supports a hardstatus.

The third form specifies the contents of the hardstatus line. '%h' is used as default string, i.e. the stored hardstatus of the current window (settable via "ESC]0;<string>^G" or "ESC_<string>ESC\\") is displayed. You can customize this to any string you like including the escapes from the "STRING ESCAPES" chapter. If you leave out the argument *string*, the current string is displayed.

You can mix the second and third form by providing the string as additional argument.

height [**-w**|**-d**] [*lines* [*cols*]]

Set the display height to a specified number of lines. When no argument is given it toggles between 24 and 42 lines display. You can also specify a width if you want to change both values. The **-w** option tells screen to leave the display size unchanged and just set the window size, **-d** vice versa.

help [-c *class*]

Not really a online help, but displays a help *screen* showing you all the key bindings. The first pages list all the internal commands followed by their current bindings. Subsequent pages will display the custom commands, one command per key. Press space when you're done reading each page, or return to exit early. All other characters are ignored. If the "-c" option is given, display all bound commands for the specified command class. See also "DEFAULT KEY BINDINGS" section.

history

Usually users work with a shell that allows easy access to previous commands. For example *cs*h has the command "!!" to repeat the last command executed. *Screen* allows you to have a primitive way of re-calling "the command that started ...": You just type the first letter of that command, then hit 'C-a {' and *screen* tries to find a previous line that matches with the 'prompt character' to the left of the cursor. This line is pasted into this window's input queue. Thus you have a crude command history (made up by the visible window and its scrollbar buffer).

hstatus *status*

Change the window's hardstatus line to the string *status*.

idle [*timeout* [*cmd args*]]

Sets a command that is run after the specified number of seconds inactivity is reached. This command will normally be the "blinker" command to create a screen blanker, but it can be any screen command. If no command is specified, only the timeout is set. A timeout of zero (or the special timeout **off**) disables the timer. If no arguments are given, the current settings are displayed.

ignorecase [**on**|**off**]

Tell screen to ignore the case of characters in searches. Default is 'off'.

info

Uses the message line to display some information about the current window: the cursor position in the form "(column,row)" starting with "(1,1)", the terminal width and height plus the size of the scrollbar buffer in lines, like in "(80,24)+50", the current state of window XON/XOFF flow control is shown like this (See also section FLOW CONTROL):

```
+flow  automatic flow control, currently on.
-flow  automatic flow control, currently off.
+(+)flow flow control enabled. Agrees with automatic control.
-(-)flow flow control disabled. Disagrees with automatic control.
+(-)flow flow control enabled. Disagrees with automatic control.
-(-)flow flow control disabled. Agrees with automatic control.
```

The current line wrap setting ('+wrap' indicates enabled, '-wrap' not) is also shown. The flags 'ins', 'org', 'app', 'log', 'mon' or 'nored' are displayed when the window is in insert mode, origin mode, application-keypad mode, has output logging, activity monitoring or partial redraw enabled.

The currently active character set (*G0*, *G1*, *G2*, or *G3*) and in square brackets the terminal character sets that are currently designated as *G0* through *G3* is shown. If the window is in UTF-8 mode, the string "UTF-8" is shown instead.

Additional modes depending on the type of the window are displayed at the end of the status line (See also chapter "WINDOW TYPES").

If the state machine of the terminal emulator is in a non-default state, the info line is started with a string identifying the current state.

For system information use the “time” command.

ins_reg [*key*]

No longer exists, use “paste” instead.

kill

Kill current window.

If there is an ‘exec’ command running then it is killed. Otherwise the process (shell) running in the window receives a HANGUP condition, the window structure is removed and *screen* (your display) switches to another window. When the last window is destroyed, *screen* exits. After a kill *screen* switches to the previously displayed window.

Note: *Emacs* users should keep this command in mind, when killing a line. It is recommended not to use “C-a” as the *screen* escape key or to rebind kill to “C-a K”.

lastmsg

Redisplay the last contents of the message/status line. Useful if you’re typing when a message appears, because the message goes away when you press a key (unless your terminal has a hardware status line). Refer to the commands “msgwait” and “msgminwait” for fine tuning.

license

Display the disclaimer page. This is done whenever *screen* is started without options, which should be often enough. See also the “startup_message” command.

lockscreen

Lock this display. Call a screenlock program (/local/bin/lck or /usr/bin/lock or a builtin if no other is available). Screen does not accept any command keys until this program terminates. Meanwhile processes in the windows may continue, as the windows are in the ‘detached’ state. The screenlock program may be changed through the environment variable \$LOCKPRG (which must be set in the shell from which *screen* is started) and is executed with the user’s uid and gid.

Warning: When you leave other shells unlocked and you have no password set on *screen*, the lock is void: One could easily re-attach from an unlocked shell. This feature should rather be called ‘lockterminal’.

log [on|off]

Start/stop writing output of the current window to a file “screenlog.*n*” in the window’s default directory, where *n* is the number of the current window. This filename can be changed with the ‘logfile’ command. If no parameter is given, the state of logging is toggled. The session log is appended to the previous contents of the file if it already exists. The current contents and the contents of the scrollback history are not included in the session log. Default is ‘off’.

logfile *filename***logfile flush** *secs*

Defines the name the logfiles will get. The default is “screenlog.%*n*”. The second form changes the number of seconds *screen* will wait before flushing the logfile buffer to the file-system. The default value is 10 seconds.

login [on|off]

Adds or removes the entry in the utmp database file for the current window. This controls if the window is ‘logged in’. When no parameter is given, the login state of the window is toggled. Additionally to that toggle, it is convenient having a ‘log in’ and a ‘log out’ key. E. g. ‘bind I login on’ and ‘bind O login off’ will map these keys to be C-a I and C-a O. The default setting (in config.h.in) should be “on” for a *screen* that runs under suid-root. Use the “deflogin” command to change the default login state for new windows.

Both commands are only present when *screen* has been compiled with utmp support.

logtstamp [on|off]

logtstamp after [secs]

logtstamp string [string]

This command controls logfile time-stamp mechanism of *screen*. If time-stamps are turned “on”, *screen* adds a string containing the current time to the logfile after two minutes of inactivity. When output continues and more than another two minutes have passed, a second time-stamp is added to document the restart of the output. You can change this timeout with the second form of the command. The third form is used for customizing the time-stamp string (‘-- %n:%t -- time-stamp -- %M/%d/%y %c:%s --\n’ by default).

mapdefault

Tell *screen* that the next input character should only be looked up in the default bindkey table. See also “bindkey”.

mapnotnext

Like mapdefault, but don’t even look in the default bindkey table.

maptimeout [timo]

Set the inter-character timer for input sequence detection to a timeout of *timo* ms. The default timeout is 300ms. Maptimeout with no arguments shows the current setting. See also “bindkey”.

markkeys string

This is a method of changing the keymap used for copy/history mode. The string is made up of *old-char=newchar* pairs which are separated by ‘:’. Example: The string “B=^B:F=^F” will change the keys ‘C-b’ and ‘C-f’ to the vi style binding (scroll up/down fill page). This happens to be the default binding for ‘B’ and ‘F’. The command “markkeys h=^B:l=^F:\$=^E” would set the mode for an emacs-style binding. If your terminal sends characters, that cause you to abort copy mode, then this command may help by binding these characters to do nothing. The no-op character is ‘@’ and is used like this: “markkeys @=L=H” if you do not want to use the ‘H’ or ‘L’ commands any longer. As shown in this example, multiple keys can be assigned to one function in a single statement.

maxwin num

Set the maximum window number *screen* will create. Doesn’t affect already existing windows. The number may only be decreased.

meta

Insert the command character (C-a) in the current window’s input stream.

monitor [on|off]

Toggles activity monitoring of windows. When monitoring is turned on and an affected window is switched into the background, you will receive the activity notification message in the status line at the first sign of output and the window will also be marked with an ‘@’ in the window-status display. Monitoring is initially off for all windows.

msgminwait sec

Defines the time *screen* delays a new message when one message is currently displayed. The default is 1 second.

msgwait *sec*

Defines the time a message is displayed if *screen* is not disturbed by other activity. The default is 5 seconds.

multiuser *on|off*

Switch between singleuser and multiuser mode. Standard *screen* operation is singleuser. In multiuser mode the commands ‘acladd’, ‘aclchg’, ‘aclgrp’ and ‘acldel’ can be used to enable (and disable) other users accessing this *screen* session.

nethack *on|off*

Changes the kind of error messages used by *screen*. When you are familiar with the game “nethack”, you may enjoy the nethack-style messages which will often blur the facts a little, but are much funnier to read. Anyway, standard messages often tend to be unclear as well.

This option is only available if *screen* was compiled with the NETHACK flag defined. The default setting is then determined by the presence of the environment variable \$NETHACKOPTIONS.

next

Switch to the next window. This command can be used repeatedly to cycle through the list of windows.

nonblock [*on|off|numsecs*]

Tell *screen* how to deal with user interfaces (displays) that cease to accept output. This can happen if a user presses ^S or a TCP/modem connection gets cut but no hangup is received. If nonblock is **off** (this is the default) *screen* waits until the display restarts to accept the output. If nonblock is **on**, *screen* waits until the timeout is reached (**on** is treated as 1s). If the display still doesn’t receive characters, *screen* will consider it “blocked” and stop sending characters to it. If at some time it restarts to accept characters, *screen* will unblock the display and redisplay the updated window contents.

number [*n*]

Change the current windows number. If the given number *n* is already used by another window, both windows exchange their numbers. If no argument is specified, the current window number (and title) is shown.

obuflimit [*limit*]

If the output buffer contains more bytes than the specified limit, no more data will be read from the windows. The default value is 256. If you have a fast display (like *xterm*), you can set it to some higher value. If no argument is specified, the current setting is displayed.

only

Kill all regions but the current one.

other

Switch to the window displayed previously. If this window does no longer exist, *other* has the same effect as *next*.

partial *on|off*

Defines whether the display should be refreshed (as with *redisplay*) after switching to the current window. This command only affects the current window. To immediately affect all windows use the *allpartial* command. Default is ‘off’, of course. This default is fixed, as there is currently no *defpartial* command.

password [*encrypted_pw*]

Present a crypted password in your “.screenrc” file and *screen* will ask for it, whenever someone attempts to resume a detached. This is useful if you have privileged programs running under *screen* and you want to

protect your session from reattach attempts by another user masquerading as your uid (i.e. any superuser.) If no crypted password is specified, *screen* prompts twice for typing a password and places its encryption in the paste buffer. Default is 'none', this disables password checking.

paste [*registers* [*dest_reg*]]

Write the (concatenated) contents of the specified registers to the stdin queue of the current window. The register '.' is treated as the paste buffer. If no parameter is given the user is prompted for a single register to paste. The paste buffer can be filled with the *copy*, *history* and *readbuf* commands. Other registers can be filled with the *register*, *readreg* and *paste* commands. If *paste* is called with a second argument, the contents of the specified registers is pasted into the named destination register rather than the window. If '.' is used as the second argument, the displays paste buffer is the destination. Note, that "paste" uses a wide variety of resources: Whenever a second argument is specified no current window is needed. When the source specification only contains registers (not the paste buffer) then there need not be a current display (terminal attached), as the registers are a global resource. The paste buffer exists once for every user.

pastefont [*on|off*]

Tell *screen* to include font information in the paste buffer. The default is not to do so. This command is especially useful for multi character fonts like kanji.

pow_break

Reopen the window's terminal line and send a break condition. See 'break'.

pow_detach

Power detach. Mainly the same as *detach*, but also sends a HANGUP signal to the parent process of *screen*. CAUTION: This will result in a logout, when *screen* was started from your login shell.

pow_detach_msg [*message*]

The *message* specified here is output whenever a 'Power detach' was performed. It may be used as a replacement for a logout message or to reset baud rate, etc. Without parameter, the current message is shown.

prev

Switch to the window with the next lower number. This command can be used repeatedly to cycle through the list of windows.

printcmd [*cmd*]

If *cmd* is not an empty string, *screen* will not use the terminal capabilities "po/pf" if it detects an ansi print sequence ESC [5 i, but pipe the output into *cmd*. This should normally be a command like "lpr" or "'cat > /tmp/scrprint'". **printcmd** without a command displays the current setting. The ansi sequence ESC \ ends printing and closes the pipe.

Warning: Be careful with this command! If other user have write access to your terminal, they will be able to fire off print commands.

process [*key*]

Stuff the contents of the specified register into *screen*'s input queue. If no argument is given you are prompted for a register name. The text is parsed as if it had been typed in from the user's keyboard. This command can be used to bind multiple actions to a single key.

quit

Kill all windows and terminate *screen*. Note that on VT100-style terminals the keys C-4 and C-\ are identical. This makes the default bindings dangerous: Be careful not to type C-a C-4 when selecting window no. 4. Use the empty bind command (as in “bind ``\””) to remove a key binding.

readbuf [-e *encoding*] [*filename*]

Reads the contents of the specified file into the paste buffer. You can tell screen the encoding of the file via the -e option. If no file is specified, the screen-exchange filename is used. See also “bufferfile” command.

readreg [-e *encoding*] [*register* [*filename*]]

Does one of two things, dependent on number of arguments: with zero or one arguments it duplicates the paste buffer contents into the register specified or entered at the prompt. With two arguments it reads the contents of the named file into the register, just as *readbuf* reads the screen-exchange file into the paste buffer. You can tell screen the encoding of the file via the -e option. The following example will paste the system’s password file into the *screen* window (using register p, where a copy remains):

```
C-a : readreg p /etc/passwd
C-a : paste p
```

redisplay

Redisplay the current window. Needed to get a full redisplay when in partial redraw mode.

register [-e *encoding*] *key string*

Save the specified *string* to the register *key*. The encoding of the string can be specified via the -e option. See also the “paste” command.

remove

Kill the current region. This is a no-op if there is only one region.

removebuf

Unlinks the screen-exchange file used by the commands “writebuf” and “readbuf”.

reset

Reset the virtual terminal to its “power-on” values. Useful when strange settings (like scroll regions or graphics character set) are left over from an application.

resize

Resize the current region. The space will be removed from or added to the region below or if there’s not enough space from the region above.

```
resize +N    increase current region height by N
resize -N    decrease current region height by N
resize N     set current region height to N
resize =     make all windows equally high
resize max   maximize current region height
resize min   minimize current region height
```

screen [-opts] [*n*] [*cmd* [*args*]]

Establish a new window. The flow-control options (-f, -fn and -fa), title (a.k.a.) option (-t), login options (-l and -ln), terminal type option (-T <term>), the all-capability-flag (-a) and scrollbar option (-h

`<num>`) may be specified with each command. The option **(-M)** turns monitoring on for this window. The option **(-L)** turns output logging on for this window. If an optional number *n* in the range 0..9 is given, the window number *n* is assigned to the newly created window (or, if this number is already in-use, the next available number). If a command is specified after “screen”, this command (with the given arguments) is started in the window; otherwise, a shell is created. Thus, if your “.screenrc” contains the lines

```
# example for .screenrc:
screen 1
screen -fn -t foobar -L 2 telnet foobar
```

screen creates a shell window (in window #1) and a window with a TELNET connection to the machine foobar (with no flow-control using the title “foobar” in window #2) and will write a logfile (“screenlog.2”) of the telnet session. Note, that unlike previous versions of *screen* no additional default window is created when “screen” commands are included in your “.screenrc” file. When the initialization is completed, *screen* switches to the last window specified in your .screenrc file or, if none, opens a default window #0. *Screen* has built in some functionality of “cu” and “telnet”. See also chapter “WINDOW TYPES”.

scrollback *num*

Set the size of the scrollbar buffer for the current windows to *num* lines. The default scrollbar is 100 lines. See also the “defscrollback” command and use “C-a i” to view the current setting.

select [*WindowID*]

Switch to the window identified by *WindowID*. This can be a prefix of a window title (alphanumeric window name) or a window number. The parameter is optional and if omitted, you get prompted for an identifier. When a new window is established, the first available number is assigned to this window. Thus, the first window can be activated by “select 0”. The number of windows is limited at compile-time by the MAXWIN configuration parameter. There are two special WindowIDs, “-” selects the internal blank window and “.” selects the current window. The latter is useful if used with screen’s “-X” option.

sessionname [*name*]

Rename the current session. Note, that for “screen -list” the name shows up with the process-id prepended. If the argument “name” is omitted, the name of this session is displayed. Caution: The \$STY environment variables still reflects the old name. This may result in confusion. The default is constructed from the tty and host names.

setenv [*var* [*string*]]

Set the environment variable *var* to value *string*. If only *var* is specified, the user will be prompted to enter a value. If no parameters are specified, the user will be prompted for both variable and value. The environment is inherited by all subsequently forked shells.

setsid [**on**|**off**]

Normally screen uses different sessions and process groups for the windows. If setsid is turned *off*, this is not done anymore and all windows will be in the same process group as the screen backend process. This also breaks job-control, so be careful. The default is *on*, of course. This command is probably useful only in rare circumstances.

shell *command*

Set the command to be used to create a new shell. This overrides the value of the environment variable \$SHELL. This is useful if you’d like to run a tty-enhancer which is expecting to execute the program specified in \$SHELL. If the command begins with a ‘-’ character, the shell will be started as a login-shell.

shelltitle *title*

Set the title for all shells created during startup or by the C-A C-c command. For details about what a title is, see the discussion entitled “TITLES (naming windows)”.

silence [**on|off**]*sec*

Toggles silence monitoring of windows. When silence is turned on and an affected window is switched into the background, you will receive the silence notification message in the status line after a specified period of inactivity (silence). The default timeout can be changed with the ‘silencewait’ command or by specifying a number of seconds instead of ‘on’ or ‘off’. Silence is initially off for all windows.

silencewait *sec*

Define the time that all windows monitored for silence should wait before displaying a message. Default 30 seconds.

sleep *num*

This command will pause the execution of a .screenrc file for *num* seconds. Keyboard activity will end the sleep. It may be used to give users a chance to read the messages output by “echo”.

slowpaste *msec*

Define the speed at which text is inserted into the current window by the paste ("C-a J") command. If the slowpaste value is nonzero text is written character by character. *screen* will make a pause of *msec* milliseconds after each single character write to allow the application to process its input. Only use slowpaste if your underlying system exposes flow control problems while pasting large amounts of text.

source *file*

Read and execute commands from file *file*. Source commands may be nested to a maximum recursion level of ten. If file is not an absolute path and screen is already processing a source command, the parent directory of the running source command file is used to search for the new command file before screen’s current directory.

Note that termcap/terminfo/termcapinfo commands only work at startup and reattach time, so they must be reached via the default screenrc files to have an effect.

sorendition [*attr* [*color*]]

Change the way *screen* does highlighting for text marking and printing messages. See the “STRING ESCAPES” chapter for the syntax of the modifiers. The default is currently “=s dd” (standout, default colors).

split

Split the current region into two new ones. All regions on the display are resized to make room for the new region. The blank window is displayed on the new region. Use the “remove” or the “only” command to delete regions.

startup_message **on|off**

Select whether you want to see the copyright notice during startup. Default is ‘on’, as you probably noticed.

stuff *string*

Stuff the string *string* in the input buffer of the current window. This is like the “paste” command but with much less overhead. You cannot paste large buffers with the “stuff” command. It is most useful for key

bindings. See also “bindkey”.

su [username [password [password2]]

Substitute the user of a display. The command prompts for all parameters that are omitted. If passwords are specified as parameters, they have to be specified un-crypted. The first password is matched against the systems passwd database, the second password is matched against the *screen* password as set with the commands “acladd” or “password”. “Su” may be useful for the *screen* administrator to test multiuser setups. When the identification fails, the user has access to the commands available for user **nobody**. These are “detach”, “license”, “version”, “help” and “displays”.

suspend

Suspend *screen*. The windows are in the ‘detached’ state, while *screen* is suspended. This feature relies on the shell being able to do job control.

term *term*

In each window’s environment *screen* opens, the \$TERM variable is set to “screen” by default. But when no description for “screen” is installed in the local termcap or terminfo data base, you set \$TERM to – say – “vt100”. This won’t do much harm, as *screen* is VT100/ANSI compatible. The use of the “term” command is discouraged for non-default purpose. That is, one may want to specify special \$TERM settings (e.g. vt100) for the next “screen rlogin othermachine” command. Use the command “screen -T vt100 rlogin othermachine” rather than setting and resetting the default.

termcap *term terminal-tweaks [window-tweaks]*

terminfo *term terminal-tweaks [window-tweaks]*

termcapinfo *term terminal-tweaks [window-tweaks]*

Use this command to modify your terminal’s termcap entry without going through all the hassles involved in creating a custom termcap entry. Plus, you can optionally customize the termcap generated for the windows. You have to place these commands in one of the screenrc startup files, as they are meaningless once the terminal emulator is booted.

If your system works uses the terminfo database rather than termcap, *screen* will understand the ‘terminfo’ command, which has the same effects as the ‘termcap’ command. Two separate commands are provided, as there are subtle syntactic differences, e.g. when parameter interpolation (using ‘%’) is required. Note that termcap names of the capabilities have to be used with the ‘terminfo’ command.

In many cases, where the arguments are valid in both terminfo and termcap syntax, you can use the command ‘termcapinfo’, which is just a shorthand for a pair of ‘termcap’ and ‘terminfo’ commands with identical arguments.

The first argument specifies which terminal(s) should be affected by this definition. You can specify multiple terminal names by separating them with ‘|’s. Use ‘*’ to match all terminals and ‘vt*’ to match all terminals that begin with “vt”.

Each *tweak* argument contains one or more termcap defines (separated by ‘:’s) to be inserted at the start of the appropriate termcap entry, enhancing it or overriding existing values. The first tweak modifies your terminal’s termcap, and contains definitions that your terminal uses to perform certain functions. Specify a null string to leave this unchanged (e. g. ”). The second (optional) tweak modifies all the window termcaps, and should contain definitions that *screen* understands (see the “VIRTUAL TERMINAL” section).

Some examples:

```
termcap xterm* LP:hs@
```

Informs *screen* that all terminals that begin with ‘xterm’ have firm auto-margins that allow the last position on the screen to be updated (LP), but they don’t really have a status line (no ‘hs’ – append ‘@’ to turn entries off). Note that we assume ‘LP’ for all terminal names that start with “vt”, but only if you don’t specify a termcap command for that terminal.

```
termcap vt* LP
termcap vt102|vt220 Z0=\E[?3h;Z1=\E[?3l
```

Specifies the firm-margined ‘LP’ capability for all terminals that begin with ‘vt’, and the second line will also add the escape-sequences to switch into (Z0) and back out of (Z1) 132-character-per-line mode if this is a VT102 or VT220. (You must specify Z0 and Z1 in your termcap to use the width-changing commands.)

```
termcap vt100 "" l0=PF1:l1=PF2:l2=PF3:l3=PF4
```

This leaves your vt100 termcap alone and adds the function key labels to each window’s termcap entry.

```
termcap h19|z19 am@:im=\E@:ei=\EO dc=\E[P
```

Takes a h19 or z19 termcap and turns off auto-margins (am@) and enables the insert mode (im) and end-insert (ei) capabilities (the ‘@’ in the ‘im’ string is after the ‘=’, so it is part of the string). Having the ‘im’ and ‘ei’ definitions put into your terminal’s termcap will cause *screen* to automatically advertise the character-insert capability in each window’s termcap. Each window will also get the delete-character capability (dc) added to its termcap, which *screen* will translate into a line-update for the terminal (we’re pretending it doesn’t support character deletion).

If you would like to fully specify each window’s termcap entry, you should instead set the \$SCREENCAP variable prior to running *screen*. See the discussion on the “VIRTUAL TERMINAL” in this manual, and the termcap(5) man page for more information on termcap definitions.

time [*string*]

Uses the message line to display the time of day, the host name, and the load averages over 1, 5, and 15 minutes (if this is available on your system). For window specific information use “info”.

If a string is specified, it changes the format of the time report like it is described in the “STRING ESCAPES” chapter. Screen uses a default of “%c:%s %M %d %H%? %l%?”.

title [*windowtitle*]

Set the name of the current window to *windowtitle*. If no name is specified, *screen* prompts for one. This command was known as ‘aka’ in previous releases.

unsetenv *var*

Unset an environment variable.

utf8 [**on**|**off** [**on**|**off**]]

Change the encoding used in the current window. If utf8 is enabled, the strings sent to the window will be UTF-8 encoded and vice versa. Omitting the parameter toggles the setting. If a second parameter is given, the display’s encoding is also changed (this should rather be done with screen’s “-U” option). See also “defutf8”, which changes the default setting of a new window.

vbell [**on**|**off**]

Sets the visual bell setting for this window. Omitting the parameter toggles the setting. If vbell is switched on, but your terminal does not support a visual bell, a ‘vbell-message’ is displayed in the status line when the bell character (^G) is received. Visual bell support of a terminal is defined by the termcap variable ‘vb’ (terminfo: ‘flash’).

Per default, vbell is off, thus the audible bell is used. See also ‘bell_msg’.

vbell_msg [*message*]

Sets the visual bell message. *message* is printed to the status line if the window receives a bell character (^G), vbell is set to “on”, but the terminal does not support a visual bell. The default message is “Wuff, Wuff!!!”. Without parameter, the current message is shown.

vbellwait *sec*

Define a delay in seconds after each display of *screen*'s visual bell message. The default is 1 second.

verbose [**on**|**off**]

If verbose is switched on, the command name is echoed, whenever a window is created (or resurrected from zombie state). Default is off. Without parameter, the current setting is shown.

version

Print the current version and the compile date in the status line.

wall *message*

Write a message to all displays. The message will appear in the terminal's status line.

width [**-w**|-**d**] [*cols* [*lines*]]

Toggle the window width between 80 and 132 columns or set it to *cols* columns if an argument is specified. This requires a capable terminal and the termcap entries "Z0" and "Z1". See the "termcap" command for more information. You can also specify a new height if you want to change both values. The **-w** option tells screen to leave the display size unchanged and just set the window size, **-d** vice versa.

windowlist [**-b**] [**-m**]**windowlist string** [*string*]**windowlist title** [*title*]

Display all windows in a table for visual window selection. The desired window can be selected via the standard movement keys (see the "copy" command) and activated via the return key. If the **-b** option is given, screen will switch to the blank window before presenting the list, so that the current window is also selectable. The **-m** option changes the order of the windows, instead of sorting by window numbers screen uses its internal most-recently-used list.

The table format can be changed with the **string** and **title** option, the title is displayed as table heading, while the lines are made by using the string setting. The default setting is "Num Name%=Flags" for the title and "%3n %t%=%f" for the lines. See the "STRING ESCAPES" chapter for more codes (e.g. color settings).

windows

Uses the message line to display a list of all the windows. Each window is listed by number with the name of process that has been started in the window (or its title); the current window is marked with a '*'; the previous window is marked with a '-'; all the windows that are "logged in" are marked with a '\$'; a background window that has received a bell is marked with a '!'; a background window that is being monitored and has had activity occur is marked with an '@'; a window which has output logging turned on is marked with '(L)'; windows occupied by other users are marked with '&'; windows in the zombie state are marked with 'Z'. If this list is too long to fit on the terminal's status line only the portion around the current window is displayed.

wrap [**on**|**off**]

Sets the line-wrap setting for the current window. When line-wrap is on, the second consecutive printable character output at the last column of a line will wrap to the start of the following line. As an added feature, backspace (^H) will also wrap through the left margin to the previous line. Default is 'on'.

writebuf [-e *encoding*] [*filename*]

Writes the contents of the paste buffer to the specified file, or the public accessible screen-exchange file if no filename is given. This is thought of as a primitive means of communication between *screen* users on the same host. If an encoding is specified the paste buffer is recoded on the fly to match the encoding. The filename can be set with the *bufferfile* command and defaults to “/tmp/screen-exchange”.

writelock [on|off|auto]

In addition to access control lists, not all users may be able to write to the same window at once. Per default, writelock is in ‘auto’ mode and grants exclusive input permission to the user who is the first to switch to the particular window. When he leaves the window, other users may obtain the writelock (automatically). The writelock of the current window is disabled by the command “writelock off”. If the user issues the command “writelock on” he keeps the exclusive write permission while switching to other windows.

xoff**xon**

Insert a CTRL-s / CTRL-q character to the stdin queue of the current window.

zmodem [off|auto|catch|pass]**zmodem sendcmd** [*string*]**zmodem recvcmd** [*string*]

Define zmodem support for screen. Screen understands two different modes when it detects a zmodem request: “pass” and “catch”. If the mode is set to “pass”, screen will relay all data to the attacher until the end of the transmission is reached. In “catch” mode screen acts as a zmodem endpoint and starts the corresponding rz/sz commands. If the mode is set to “auto”, screen will use “catch” if the window is a tty (e.g. a serial line), otherwise it will use “pass”.

You can define the templates screen uses in “catch” mode via the second and the third form.

Note also that this is an experimental feature.

zombie [*keys*]**defzombie** [*keys*]

Per default *screen* windows are removed from the window list as soon as the windows process (e.g. shell) exits. When a string of two keys is specified to the zombie command, ‘dead’ windows will remain in the list. The **kill** command may be used to remove such a window. Pressing the first key in the dead window has the same effect. When pressing the second key, *screen* will attempt to resurrect the window. The process that was initially running in the window will be launched again. Calling **zombie** without parameters will clear the zombie setting, thus making windows disappear when their process exits.

As the zombie-setting is manipulated globally for all windows, this command should only be called **defzombie**. Until we need this as a per window setting, the commands **zombie** and **defzombie** are synonymous.

THE MESSAGE LINE

Screen displays informational messages and other diagnostics in a *message line*. While this line is distributed to appear at the bottom of the screen, it can be defined to appear at the top of the screen during compilation. If your terminal has a status line defined in its termcap, *screen* will use this for displaying its messages, otherwise a line of the current screen will be temporarily overwritten and output will be momentarily interrupted. The message line is automatically removed after a few seconds delay, but it can also be removed early (on terminals without a status line) by beginning to type.

The message line facility can be used by an application running in the current window by means of the ANSI *Privacy message* control sequence. For instance, from within the shell, try something like:


```
echo ^<esc>^Hello world from window '$WINDOW' <esc>\\
```

where '`<esc>`' is an *escape*, '^' is a literal up-arrow, and '\\ turns into a single backslash.

WINDOW TYPES

Screen provides three different window types. New windows are created with *screen*'s **screen** command (see also the entry in chapter "CUSTOMIZATION"). The first parameter to the **screen** command defines which type of window is created. The different window types are all special cases of the normal type. They have been added in order to allow *screen* to be used efficiently as a console multiplexer with 100 or more windows.

- The normal window contains a shell (default, if no parameter is given) or any other system command that could be executed from a shell (e.g. **slogin**, etc...)
- If a tty (character special device) name (e.g. `"/dev/ttya"`) is specified as the first parameter, then the window is directly connected to this device. This window type is similar to `"screen cu -l /dev/ttya"`. Read and write access is required on the device node, an exclusive open is attempted on the node to mark the connection line as busy. An optional parameter is allowed consisting of a comma separated list of flags in the notation used by `stty(1)`:

`<baud_rate>`

Usually 300, 1200, 9600 or 19200. This affects transmission as well as receive speed.

`cs8` or `cs7`

Specify the transmission of eight (or seven) bits per byte.

`ixon` or `-ixon`

Enables (or disables) software flow-control (CTRL-S/CTRL-Q) for sending data.

`ixoff` or `-ixon`

Enables (or disables) software flow-control for receiving data.

`istrip` or `-istrip`

Clear (or keep) the eight bit in each received byte.

You may want to specify as many of these options as applicable. Unspecified options cause the terminal driver to make up the parameter values of the connection. These values are system dependant and may be in defaults or values saved from a previous connection.

For tty windows, the **info** command shows some of the modem control lines in the status line. These may include 'RTS', 'CTS', 'DTR', 'DSR', 'CD' and more. This depends on the available `ioctl()`'s and system header files as well as the on the physical capabilities of the serial board. Signals that are logical low (inactive) have their name preceded by an exclamation mark (!), otherwise the signal is logical high (active). Signals not supported by the hardware but available to the `ioctl()` interface are usually shown low.

When the CLOCAL status bit is true, the whole set of modem signals is placed inside curly braces ({ and }). When the CRTSCTS or TIOCSOFTCAR bit is set, the signals 'CTS' or 'CD' are shown in parenthesis, respectively.

For tty windows, the command **break** causes the Data transmission line (TxD) to go low for a specified period of time. This is expected to be interpreted as break signal on the other side. No data is sent and no modem control line is changed when a **break** is issued.

- If the first parameter is `"/telnet"`, the second parameter is expected to be a host name, and an optional third parameter may specify a TCP port number (default decimal 23). Screen will connect to a server listening on the remote host and use the telnet protocol to communicate with that server. For telnet windows, the command **info** shows details about the connection in square brackets ([and]) at the end of the status line.

- b BINARY. The connection is in binary mode.
- e ECHO. Local echo is disabled.
- c SGA. The connection is in 'character mode' (default: 'line mode').
- t TTYPE. The terminal type has been requested by the remote host. Screen sends the name "screen" unless instructed otherwise (see also the command 'term').
- w NAWS. The remote site is notified about window size changes.
- f LFLOW. The remote host will send flow control information. (Ignored at the moment.)

Additional flags for debugging are x, t and n (XDISPLOC, TSPEED and NEWENV).

For telnet windows, the command **break** sends the telnet code IAC BREAK (decimal 243) to the remote host.

This window type is only available if *screen* was compiled with the BUILTIN_TELNET option defined.

STRING ESCAPES

Screen provides an escape mechanism to insert information like the current time into messages or file names. The escape character is '%' with one exception: inside of a window's hardstatus '^%' ('^E') is used instead.

Here is the full list of supported escapes:

- % the escape character itself
- a either 'am' or 'pm'
- A either 'AM' or 'PM'
- c current time HH:MM in 24h format
- C current time HH:MM in 12h format
- d day number
- D weekday name
- f flags of the window
- F sets %? to true if the window has the focus
- h hardstatus of the window
- H hostname of the system
- l current load of the system
- m month number
- M month name
- n window number
- s seconds
- t window title
- u all other users on this window
- w all window numbers and names. With '-' qualifier: up to the current window; with '+' qualifier: starting with the window after the current one.
- W all window numbers and names except the current one

| | |
|---|---|
| y | last two digits of the year number |
| Y | full year number |
| ? | the part to the next '%?' is displayed only if a '%' escape inside the part expands to a non-empty string |
| : | else part of '%?' |
| = | pad the string to the display's width (like TeX's hfill). If a number is specified, pad to the percentage of the window's width. A '0' qualifier tells screen to treat the number as absolute position. You can specify to pad relative to the last absolute pad position by adding a '+' qualifier or to pad relative to the right margin by using '-'. The padding truncates the string if the specified position lies before the current position. Add the 'L' qualifier to change this. |
| < | same as '%=' but just do truncation, do not fill with spaces |
| > | mark the current text position for the next truncation. When screen needs to do truncation, it tries to do it in a way that the marked position gets moved to the specified percentage of the output area. (The area starts from the last absolute pad position and ends with the position specified by the truncation operator.) The 'L' qualifier tells screen to mark the truncated parts with '...'. |
| { | attribute/color modifier string terminated by the next "}" |
| ` | Substitute with the output of a 'backtick' command. The length qualifier is misused to identify one of the commands. |

The 'c' and 'C' escape may be qualified with a '0' to make *screen* use zero instead of space as fill character. The '0' qualifier also makes the '=' escape use absolute positions. The 'n' and '=' escapes understand a length qualifier (e.g. '%3n'), 'D' and 'M' can be prefixed with 'L' to generate long names, 'w' and 'W' also show the window flags if 'L' is given.

An attribute/color modifier is used to change the attributes or the color settings. Its format is "[attribute modifier] [color description]". The attribute modifier must be prefixed by a change type indicator if it can be confused with a color description. The following change types are known:

| | |
|---|--|
| + | add the specified set to the current attributes |
| - | remove the set from the current attributes |
| ! | invert the set in the current attributes |
| = | change the current attributes to the specified set |

The attribute set can either be specified as a hexadecimal number or a combination of the following letters:

| | |
|---|-----------|
| d | dim |
| u | underline |
| b | bold |
| r | reverse |
| s | standout |
| B | blinking |

Colors are coded either as a hexadecimal number or two letters specifying the desired background and foreground color (in that order). The following colors are known:

| | |
|---|---------|
| k | black |
| r | red |
| g | green |
| y | yellow |
| b | blue |
| m | magenta |
| c | cyan |
| w | white |

d default color
 . leave color unchanged

The capitalized versions of the letter specify bright colors. You can also use the pseudo-color 'i' to set just the brightness and leave the color unchanged.

A one digit/letter color description is treated as foreground or background color dependant on the current attributes: if reverse mode is set, the background color is changed instead of the foreground color. If you don't like this, prefix the color with a ".". If you want the same behaviour for two-letter color descriptions, also prefix them with a ".".

As a special case, "%{-}" restores the attributes and colors that were set before the last change was made (i.e. pops one level of the color-change stack).

Examples:

"G" set color to bright green
 "+b r" use bold red
 "= yd" clear all attributes, write in default color on yellow background.

%-Lw%{= BW}%50>%n%f* %t%{-}%+Lw%<

The available windows centered at the current window and truncated to the available width. The current window is displayed white on blue. This can be used with "hardstatus alwayslastline".

%?%F%{.R.}%?%3n %t%? [%h]%?

The window number and title and the window's hardstatus, if one is set. Also use a red background if this is the active focus. Useful for "caption string".

FLOW-CONTROL

Each window has a flow-control setting that determines how *screen* deals with the XON and XOFF characters (and perhaps the interrupt character). When flow-control is turned off, *screen* ignores the XON and XOFF characters, which allows the user to send them to the current program by simply typing them (useful for the *emacs* editor, for instance). The trade-off is that it will take longer for output from a "normal" program to pause in response to an XOFF. With flow-control turned on, XON and XOFF characters are used to immediately pause the output of the current window. You can still send these characters to the current program, but you must use the appropriate two-character *screen* commands (typically "C-a q" (xon) and "C-a s" (xoff)). The xon/xoff commands are also useful for typing C-s and C-q past a terminal that intercepts these characters.

Each window has an initial flow-control value set with either the `-f` option or the "defflow" .screenrc command. Per default the windows are set to automatic flow-switching. It can then be toggled between the three states 'fixed on', 'fixed off' and 'automatic' interactively with the "flow" command bound to "C-a f".

The automatic flow-switching mode deals with flow control using the TIOCPKT mode (like "rlogin" does). If the tty driver does not support TIOCPKT, *screen* tries to find out the right mode based on the current setting of the application keypad – when it is enabled, flow-control is turned off and visa versa. Of course, you can still manipulate flow-control manually when needed.

If you're running with flow-control enabled and find that pressing the interrupt key (usually C-c) does not interrupt the display until another 6-8 lines have scrolled by, try running *screen* with the "interrupt" option (add the "interrupt" flag to the "flow" command in your .screenrc, or use the `-i` command-line option). This causes the output that *screen* has accumulated from the interrupted program to be flushed. One disadvantage is that the virtual terminal's memory contains the non-flushed version of the output, which in rare cases can cause minor inaccuracies in the output. For example, if you switch screens and return, or update the screen with "C-a l" you would see the version of the output you would have gotten without "interrupt" being on. Also, you might need to turn off flow-control (or use auto-flow mode to turn it off automatically) when running a program that expects you to type the interrupt character as input, as it is possible to interrupt the output of the virtual terminal to your physical terminal when flow-control is enabled. If this happens, a simple refresh of the screen with "C-a l" will restore it. Give each mode a try, and use whichever mode you find more comfortable.

TITLES (naming windows)

You can customize each window's name in the window display (viewed with the "windows" command (C-a w)) by setting it with one of the title commands. Normally the name displayed is the actual command name of the program created in the window. However, it is sometimes useful to distinguish various programs of the same name or to change the name on-the-fly to reflect the current state of the window.

The default name for all shell windows can be set with the "shelltitle" command in the .screenrc file, while all other windows are created with a "screen" command and thus can have their name set with the `-t` option. Interactively, there is the title-string escape-sequence (`<esc>kname<esc>\`) and the "title" command (C-a A). The former can be output from an application to control the window's name under software control, and the latter will prompt for a name when typed. You can also bind pre-defined names to keys with the "title" command to set things quickly without prompting.

Finally, *screen* has a shell-specific heuristic that is enabled by setting the window's name to "*search|name*" and arranging to have a null title escape-sequence output as a part of your prompt. The *search* portion specifies an end-of-prompt search string, while the *name* portion specifies the default shell name for the window. If the *name* ends in a ':' *screen* will add what it believes to be the current command running in the window to the end of the window's shell name (e. g. "*name:cmd*"). Otherwise the current command name supersedes the shell name while it is running.

Here's how it works: you must modify your shell prompt to output a null title-escape-sequence (`<esc>k<esc>\`) as a part of your prompt. The last part of your prompt must be the same as the string you specified for the *search* portion of the title. Once this is set up, *screen* will use the title-escape-sequence to clear the previous command name and get ready for the next command. Then, when a newline is received from the shell, a search is made for the end of the prompt. If found, it will grab the first word after the matched string and use it as the command name. If the command name begins with either '!', '%', or '^' *screen* will use the first word on the following line (if found) in preference to the just-found name. This helps csh users get better command names when using job control or history recall commands.

Here's some .screenrc examples:

```
screen -t top 2 nice top
```

Adding this line to your .screenrc would start a nice-d version of the "top" command in window 2 named "top" rather than "nice".

```
shelltitle '> |csh'
screen 1
```

These commands would start a shell with the given shelltitle. The title specified is an auto-title that would expect the prompt and the typed command to look something like the following:

```
/usr/joe/src/dir> trn
```

(it looks after the '>' for the command name). The window status would show the name "trn" while the command was running, and revert to "csh" upon completion.

```
bind R screen -t '% |root:' su
```

Having this command in your .screenrc would bind the key sequence "C-a R" to the "su" command and give it an auto-title name of "root:". For this auto-title to work, the screen could look something like this:

```
% !em
emacs file.c
```

Here the user typed the csh history command "!em" which ran the previously entered "emacs" command. The window status would show "root:emacs" during the execution of the command, and revert to simply "root:" at its completion.

```
bind o title
bind E title ""
```

```
bind u title (unknown)
```

The first binding doesn't have any arguments, so it would prompt you for a title. when you type "C-a o". The second binding would clear an auto-title's current setting (C-a E). The third binding would set the current window's title to "(unknown)" (C-a u).

One thing to keep in mind when adding a null title-escape-sequence to your prompt is that some shells (like the *csh*) count all the non-control characters as part of the prompt's length. If these invisible characters aren't a multiple of 8 then backspacing over a tab will result in an incorrect display. One way to get around this is to use a prompt like this:

```
set prompt='^[0000m^[k^[\'% '
```

The escape-sequence "<esc>[0000m" not only normalizes the character attributes, but all the zeros round the length of the invisible characters up to 8. Bash users will probably want to echo the escape sequence in the PROMPT_COMMAND:

```
PROMPT_COMMAND='echo -n -e "\033k\033\134"'
```

(I used "134" to output a '\ because of a bug in bash v1.04).

THE VIRTUAL TERMINAL

Each window in a *screen* session emulates a VT100 terminal, with some extra functions added. The VT100 emulator is hard-coded, no other terminal types can be emulated.

Usually *screen* tries to emulate as much of the VT100/ANSI standard as possible. But if your terminal lacks certain capabilities, the emulation may not be complete. In these cases *screen* has to tell the applications that some of the features are missing. This is no problem on machines using *termcap*, because *screen* can use the \$TERMCAP variable to customize the standard *screen* *termcap*.

But if you do a rlogin on another machine or your machine supports only *terminfo* this method fails. Because of this, *screen* offers a way to deal with these cases. Here is how it works:

When *screen* tries to figure out a terminal name for itself, it first looks for an entry named "screen.<term>", where <term> is the contents of your \$TERM variable. If no such entry exists, *screen* tries "screen" (or "screen-w" if the terminal is wide (132 cols or more)). If even this entry cannot be found, "vt100" is used as a substitute.

The idea is that if you have a terminal which doesn't support an important feature (e.g. delete char or clear to EOS) you can build a new *termcap*/*terminfo* entry for *screen* (named "screen.<dumbterm>") in which this capability has been disabled. If this entry is installed on your machines you are able to do a rlogin and still keep the correct *termcap*/*terminfo* entry. The terminal name is put in the \$TERM variable of all new windows. *Screen* also sets the \$TERMCAP variable reflecting the capabilities of the virtual terminal emulated. Notice that, however, on machines using the *terminfo* database this variable has no effect. Furthermore, the variable \$WINDOW is set to the window number of each window.

The actual set of capabilities supported by the virtual terminal depends on the capabilities supported by the physical terminal. If, for instance, the physical terminal does not support underscore mode, *screen* does not put the 'us' and 'ue' capabilities into the window's \$TERMCAP variable, accordingly. However, a minimum number of capabilities must be supported by a terminal in order to run *screen*; namely scrolling, clear screen, and direct cursor addressing (in addition, *screen* does not run on hardcopy terminals or on terminals that over-strike).

Also, you can customize the \$TERMCAP value used by *screen* by using the "termcap" .screenrc command, or by defining the variable \$SCREENCAP prior to startup. When the latter is defined, its value will be copied verbatim into each window's \$TERMCAP variable. This can either be the full terminal definition, or a filename where the terminal "screen" (and/or "screen-w") is defined.

Note that *screen* honors the "terminfo" .screenrc command if the system uses the *terminfo* database rather than *termcap*.

When the boolean ‘G0’ capability is present in the termcap entry for the terminal on which *screen* has been called, the terminal emulation of *screen* supports multiple character sets. This allows an application to make use of, for instance, the VT100 graphics character set or national character sets. The following control functions from ISO 2022 are supported: *lock shift G0 (SI)*, *lock shift G1 (SO)*, *lock shift G2*, *lock shift G3*, *single shift G2*, and *single shift G3*. When a virtual terminal is created or reset, the ASCII character set is designated as *G0* through *G3*. When the ‘G0’ capability is present, *screen* evaluates the capabilities ‘S0’, ‘E0’, and ‘C0’ if present. ‘S0’ is the sequence the terminal uses to enable and start the graphics character set rather than *SI*. ‘E0’ is the corresponding replacement for *SO*. ‘C0’ gives a character by character translation string that is used during semi-graphics mode. This string is built like the ‘acsc’ terminfo capability.

When the ‘po’ and ‘pf’ capabilities are present in the terminal’s termcap entry, applications running in a *screen* window can send output to the printer port of the terminal. This allows a user to have an application in one window sending output to a printer connected to the terminal, while all other windows are still active (the printer port is enabled and disabled again for each chunk of output). As a side-effect, programs running in different windows can send output to the printer simultaneously. Data sent to the printer is not displayed in the window. The *info* command displays a line starting ‘PRIN’ while the printer is active.

Screen maintains a hardstatus line for every window. If a window gets selected, the display’s hardstatus will be updated to match the window’s hardstatus line. If the display has no hardstatus the line will be displayed as a standard *screen* message. The hardstatus line can be changed with the ANSI Application Program Command (APC): “ESC_<string>ESC\”. As a convenience for xterm users the sequence “ESC]0..2;<string>^G” is also accepted.

Some capabilities are only put into the \$TERMCAP variable of the virtual terminal if they can be efficiently implemented by the physical terminal. For instance, ‘dl’ (delete line) is only put into the \$TERMCAP variable if the terminal supports either delete line itself or scrolling regions. Note that this may provoke confusion, when the session is reattached on a different terminal, as the value of \$TERMCAP cannot be modified by parent processes.

The “alternate screen” capability is not enabled by default. Set the **altscreen** .screenrc command to enable it.

The following is a list of control sequences recognized by *screen*. “(V)” and “(A)” indicate VT100-specific and ANSI- or ISO-specific functions, respectively.

| | | |
|-----------------|-----|----------------------------------|
| ESC E | | Next Line |
| ESC D | | Index |
| ESC M | | Reverse Index |
| ESC H | | Horizontal Tab Set |
| ESC Z | | Send VT100 Identification String |
| ESC 7 | (V) | Save Cursor and Attributes |
| ESC 8 | (V) | Restore Cursor and Attributes |
| ESC [s | (A) | Save Cursor and Attributes |
| ESC [u | (A) | Restore Cursor and Attributes |
| ESC c | | Reset to Initial State |
| ESC g | | Visual Bell |
| ESC Pn p | | Cursor Visibility (97801) |
| Pn = 6 | | Invisible |
| 7 | | Visible |
| ESC = | (V) | Application Keypad Mode |
| ESC > | (V) | Numeric Keypad Mode |

| | |
|----------------------------|---|
| ESC # 8 | (V) Fill Screen with E's |
| ESC \ | (A) String Terminator |
| ESC ^ | (A) Privacy Message String (Message Line) |
| ESC ! | Global Message String (Message Line) |
| ESC k | A. k. a. Definition String |
| ESC P | (A) Device Control String. Outputs a string directly to the host terminal without interpretation. |
| ESC _ | (A) Application Program Command (Hardstatus) |
| ESC] 0 ; string ^G | (A) Operating System Command (Hardstatus, xterm title hack) |
| ESC] 83 ; cmd ^G | (A) Execute screen command. This only works if multi-user support is compiled into screen. The pseudo-user “:window:” is used to check the access control list. Use “addacl :window: -rwx #?” to create a user with no rights and allow only the needed commands. |
| Control-N | (A) Lock Shift G1 (SO) |
| Control-O | (A) Lock Shift G0 (SI) |
| ESC n | (A) Lock Shift G2 |
| ESC o | (A) Lock Shift G3 |
| ESC N | (A) Single Shift G2 |
| ESC O | (A) Single Shift G3 |
| ESC (Pcs | (A) Designate character set as G0 |
| ESC) Pcs | (A) Designate character set as G1 |
| ESC * Pcs | (A) Designate character set as G2 |
| ESC + Pcs | (A) Designate character set as G3 |
| ESC [Pn ; Pn H | Direct Cursor Addressing |
| ESC [Pn ; Pn f | same as above |
| ESC [Pn J | Erase in Display |
| Pn = None or 0 | From Cursor to End of Screen |
| 1 | From Beginning of Screen to Cursor |
| 2 | Entire Screen |
| ESC [Pn K | Erase in Line |
| Pn = None or 0 | From Cursor to End of Line |
| 1 | From Beginning of Line to Cursor |
| 2 | Entire Line |
| ESC [Pn X | Erase character |
| ESC [Pn A | Cursor Up |
| ESC [Pn B | Cursor Down |
| ESC [Pn C | Cursor Right |
| ESC [Pn D | Cursor Left |
| ESC [Pn E | Cursor next line |

| | | |
|----------------------------|-----|---|
| ESC [Pn F | | Cursor previous line |
| ESC [Pn G | | Cursor horizontal position |
| ESC [Pn ‘ | | same as above |
| ESC [Pn d | | Cursor vertical position |
| ESC [Ps ;...; Ps m | | Select Graphic Rendition |
| Ps = None or 0 | | Default Rendition |
| 1 | | Bold |
| 2 | (A) | Faint |
| 3 | (A) | <i>Standout</i> Mode (ANSI: Italicized) |
| 4 | | Underlined |
| 5 | | Blinking |
| 7 | | Negative Image |
| 22 | (A) | Normal Intensity |
| 23 | (A) | <i>Standout</i> Mode off (ANSI: Italicized off) |
| 24 | (A) | Not Underlined |
| 25 | (A) | Not Blinking |
| 27 | (A) | Positive Image |
| 30 | (A) | Foreground Black |
| 31 | (A) | Foreground Red |
| 32 | (A) | Foreground Green |
| 33 | (A) | Foreground Yellow |
| 34 | (A) | Foreground Blue |
| 35 | (A) | Foreground Magenta |
| 36 | (A) | Foreground Cyan |
| 37 | (A) | Foreground White |
| 39 | (A) | Foreground Default |
| 40 | (A) | Background Black |
| ... | | |
| 49 | (A) | Background Default |
| ESC [Pn g | | Tab Clear |
| Pn = None or 0 | | Clear Tab at Current Position |
| 3 | | Clear All Tabs |
| ESC [Pn ; Pn r | (V) | Set Scrolling Region |
| ESC [Pn I | (A) | Horizontal Tab |
| ESC [Pn Z | (A) | Backward Tab |
| ESC [Pn L | (A) | Insert Line |
| ESC [Pn M | (A) | Delete Line |
| ESC [Pn @ | (A) | Insert Character |

| | | |
|----------------------------|-----|--|
| ESC [Pn P | (A) | Delete Character |
| ESC [Pn S | | Scroll Scrolling Region Up |
| ESC [Pn T | | Scroll Scrolling Region Down |
| ESC [Pn ^ | | same as above |
| ESC [Ps ;...; Ps h | | Set Mode |
| ESC [Ps ;...; Ps l | | Reset Mode |
| Ps = 4 | (A) | Insert Mode |
| 20 | (A) | <i>Automatic Linefeed</i> Mode |
| 34 | | Normal Cursor Visibility |
| ?1 | (V) | Application Cursor Keys |
| ?3 | (V) | Change Terminal Width to 132 columns |
| ?5 | (V) | Reverse Video |
| ?6 | (V) | <i>Origin</i> Mode |
| ?7 | (V) | <i>Wrap</i> Mode |
| ?9 | | X10 mouse tracking |
| ?25 | (V) | Visible Cursor |
| ?47 | | Alternate Screen (old xterm code) |
| ?1000 | (V) | VT200 mouse tracking |
| ?1047 | | Alternate Screen (new xterm code) |
| ?1049 | | Alternate Screen (new xterm code) |
| ESC [5 i | (A) | Start relay to printer (ANSI Media Copy) |
| ESC [4 i | (A) | Stop relay to printer (ANSI Media Copy) |
| ESC [8 ; Ph ; Pw t | | Resize the window to 'Ph' lines and 'Pw' columns (SunView special) |
| ESC [c | | Send VT100 Identification String |
| ESC [x | | Send Terminal Parameter Report |
| ESC [> c | | Send VT220 Secondary Device Attributes String |
| ESC [6 n | | Send Cursor Position Report |

INPUT TRANSLATION

In order to do a full VT100 emulation *screen* has to detect that a sequence of characters in the input stream was generated by a keypress on the user's keyboard and insert the VT100 style escape sequence. *Screen* has a very flexible way of doing this by making it possible to map arbitrary commands on arbitrary sequences of characters. For standard VT100 emulation the command will always insert a string in the input buffer of the window (see also command **stuff** in the command table). Because the sequences generated by a keypress can change after a reattach from a different terminal type, it is possible to bind commands to the termcap name of the keys. *Screen* will insert the correct binding after each reattach. See the **bindkey** command for further details on the syntax and examples.

Here is the table of the default key bindings. (A) means that the command is executed if the keyboard is switched into application mode.

| Key name | Termcap name | Command |
|-----------|--------------|--------------|
| Cursor up | ku | stuff \033[A |

| | | | |
|-----------------|----|----------------|-----|
| | | stuff \033OA | (A) |
| Cursor down | kd | stuff \033[B | |
| | | stuff \033OB | (A) |
| Cursor right | kr | stuff \033[C | |
| | | stuff \033OC | (A) |
| Cursor left | kl | stuff \033[D | |
| | | stuff \033OD | (A) |
| Function key 0 | k0 | stuff \033[10~ | |
| Function key 1 | k1 | stuff \033OP | |
| Function key 2 | k2 | stuff \033OQ | |
| Function key 3 | k3 | stuff \033OR | |
| Function key 4 | k4 | stuff \033OS | |
| Function key 5 | k5 | stuff \033[15~ | |
| Function key 6 | k6 | stuff \033[17~ | |
| Function key 7 | k7 | stuff \033[18~ | |
| Function key 8 | k8 | stuff \033[19~ | |
| Function key 9 | k9 | stuff \033[20~ | |
| Function key 10 | k; | stuff \033[21~ | |
| Function key 11 | F1 | stuff \033[23~ | |
| Function key 12 | F2 | stuff \033[24~ | |
| Home | kh | stuff \033[1~ | |
| End | kH | stuff \033[4~ | |
| Insert | kI | stuff \033[2~ | |
| Delete | kD | stuff \033[3~ | |
| Page up | kP | stuff \033[5~ | |
| Page down | kN | stuff \033[6~ | |
| Keypad 0 | f0 | stuff 0 | |
| | | stuff \033Op | (A) |
| Keypad 1 | f1 | stuff 1 | |
| | | stuff \033Oq | (A) |
| Keypad 2 | f2 | stuff 2 | |
| | | stuff \033Or | (A) |
| Keypad 3 | f3 | stuff 3 | |
| | | stuff \033Os | (A) |
| Keypad 4 | f4 | stuff 4 | |
| | | stuff \033Ot | (A) |
| Keypad 5 | f5 | stuff 5 | |
| | | stuff \033Ou | (A) |
| Keypad 6 | f6 | stuff 6 | |
| | | stuff \033Ov | (A) |
| Keypad 7 | f7 | stuff 7 | |
| | | stuff \033Ow | (A) |
| Keypad 8 | f8 | stuff 8 | |
| | | stuff \033Ox | (A) |
| Keypad 9 | f9 | stuff 9 | |
| | | stuff \033Oy | (A) |
| Keypad + | f+ | stuff + | |
| | | stuff \033Ok | (A) |
| Keypad - | f- | stuff - | |
| | | stuff \033Om | (A) |
| Keypad * | f* | stuff * | |
| | | stuff \033Oj | (A) |
| Keypad / | f/ | stuff / | |
| | | stuff \033Oo | (A) |

| | | | |
|--------------|----|--------------|-----|
| Keypad = | fq | stuff = | |
| Keypad . | f. | stuff \033OX | (A) |
| Keypad , | f, | stuff . | |
| Keypad enter | fe | stuff \033On | (A) |
| | | stuff , | |
| | | stuff \033OI | (A) |
| | | stuff \015 | |
| | | stuff \033OM | (A) |

SPECIAL TERMINAL CAPABILITIES

The following table describes all terminal capabilities that are recognized by *screen* and are not in the *termcap(5)* manual. You can place these capabilities in your *termcap* entries (in */etc/termcap*) or use them with the commands *'termcap'*, *'terminfo'* and *'termcapinfo'* in your *screenrc* files. It is often not possible to place these capabilities in the *terminfo* database.

| | | |
|-----------|--------|--|
| LP | (bool) | Terminal has VT100 style margins ('magic margins'). Note that this capability is obsolete because <i>screen</i> uses the standard 'xn' instead. |
| Z0 | (str) | Change width to 132 columns. |
| Z1 | (str) | Change width to 80 columns. |
| WS | (str) | Resize display. This capability has the desired width and height as arguments. <i>SunView(tm)</i> example: '\E[8;%d;%dt'. |
| NF | (bool) | Terminal doesn't need flow control. Send ^S and ^Q direct to the application. Same as 'flow off'. The opposite of this capability is 'nx'. |
| G0 | (bool) | Terminal can deal with ISO 2022 font selection sequences. |
| S0 | (str) | Switch charset 'G0' to the specified charset. Default is '\E(%.'. |
| E0 | (str) | Switch charset 'G0' back to standard charset. Default is '\E(B'. |
| C0 | (str) | Use the string as a conversion table for font '0'. See the 'ac' capability for more details. |
| CS | (str) | Switch cursor-keys to application mode. |
| CE | (str) | Switch cursor-keys back to normal mode. |
| AN | (bool) | Turn on autonuke. See the 'autonuke' command for more details. |
| OL | (num) | Set the output buffer limit. See the 'obuflimit' command for more details. |
| KJ | (str) | Set the encoding of the terminal. See the 'encoding' command for valid encodings. |
| AF | (str) | Change character foreground color in an ANSI conform way. This capability will almost always be set to '\E[3%dm' ('\E[3%p1%dm' on <i>terminfo</i> machines). |
| AB | (str) | Same as 'AF', but change background color. |
| AX | (bool) | Does understand ANSI set default fg/bg color (\E[39m / \E[49m). |
| XC | (str) | Describe a translation of characters to strings depending on the current font. More details follow in the next section. |
| XT | (bool) | Terminal understands special xterm sequences (OSC, mouse tracking). |
| C8 | (bool) | Terminal needs bold to display high-intensity colors (e.g. Eterm). |
| TF | (bool) | Add missing capabilities to the <i>termcap/info</i> entry. (Set by default). |

CHARACTER TRANSLATION

Screen has a powerful mechanism to translate characters to arbitrary strings depending on the current font and terminal type. Use this feature if you want to work with a common standard character set (say

ISO8851-latin1) even on terminals that scatter the more unusual characters over several national language font pages.

Syntax:

```
XC=<charset-mapping>{,<charset-mapping>}
<charset-mapping> := <designator><template>{,<mapping>}
<mapping> := <char-to-be-mapped><template-arg>
```

The things in braces may be repeated any number of times.

A *<charset-mapping>* tells *screen* how to map characters in font *<designator>* ('B': Ascii, 'A': UK, 'K': german, etc.) to strings. Every *<mapping>* describes to what string a single character will be translated. A template mechanism is used, as most of the time the codes have a lot in common (for example strings to switch to and from another charset). Each occurrence of '%' in *<template>* gets substituted with the *<template-arg>* specified together with the character. If your strings are not similar at all, then use '%' as a template and place the full string in *<template-arg>*. A quoting mechanism was added to make it possible to use a real '%'. The '\ ' character quotes the special characters '\', '\%', and '\,'.

Here is an example:

```
termcap hp700 'XC=B\E(K%\E(B,\304[\326\\,\334]'
```

This tells *screen* how to translate ISOlatin1 (charset 'B') upper case umlaut characters on a hp700 terminal that has a german charset. '\304' gets translated to '\E(K\E(B' and so on. Note that this line gets parsed *three* times before the internal lookup table is built, therefore a lot of quoting is needed to create a single '\ '.

Another extension was added to allow more emulation: If a mapping translates the unquoted '%' char, it will be sent to the terminal whenever *screen* switches to the corresponding *<designator>*. In this special case the template is assumed to be just '%' because the charset switch sequence and the character mappings normally haven't much in common.

This example shows one use of the extension:

```
termcap xterm 'XC=K%,%\E(B,[\304,\\326,]\334'
```

Here, a part of the german ('K') charset is emulated on an xterm. If *screen* has to change to the 'K' charset, '\E(B' will be sent to the terminal, i.e. the ASCII charset is used instead. The template is just '%', so the mapping is straightforward: '[' to '\304', '\ ' to '\326', and ']' to '\334'.

ENVIRONMENT

| | |
|----------------|--|
| COLUMNS | Number of columns on the terminal (overrides termcap entry). |
| HOME | Directory in which to look for .screenrc. |
| LINES | Number of lines on the terminal (overrides termcap entry). |
| LOCKPRG | Screen lock program. |
| NETHACKOPTIONS | Turns on nethack option. |
| PATH | Used for locating programs to run. |
| SCREENCAP | For customizing a terminal's TERMCAP value. |
| SCREENDIR | Alternate socket directory. |
| SCREENRC | Alternate user screenrc file. |
| SHELL | Default shell program for opening windows (default "/bin/sh"). |
| STY | Alternate socket name. |

SYSSCREENRC

Alternate system screenrc file.

TERM

Terminal name.

TERMCAP

Terminal description.

WINDOW

Window number of a window (at creation time).

FILES

.../screen-4.?.?/etc/screenrc

.../screen-4.?.?/etc/etcscreenrc

Examples in the *screen* distribution package for private and global initialization files.**\$SYSSCREENRC**

/etc/screenrc

screen initialization commands**\$SCREENRC**

\$HOME/.screenrc

Read in after /etc/screenrc

\$SCREENDIR/S-<login>

/local/screens/S-<login>

Socket directories (default)

/usr/tmp/screens/S-<login>

Alternate socket directories.

<socket directory>/termcap

Written by the "termcap" output function

/usr/tmp/screens/screen-exchange

or

/tmp/screen-exchange

screen 'interprocess communication buffer'

hardcopy.[0-9]

Screen images created by the hardcopy function

screenlog.[0-9]

Output log files created by the log function

/usr/lib/terminfo/?/*

or

/etc/termcap

Terminal capability databases

/etc/utmp

Login records

\$LOCKPRG

Program that locks a terminal.

SEE ALSO

termcap(5), utmp(5), vi(1), captinfo(1), tic(1)

AUTHORS

Originally created by Oliver Laumann, this latest version was produced by Wayne Davison, Juergen Weigert and Michael Schroeder.

COPYLEFT

Copyright (C) 1993-2003

Juergen Weigert (jnweiger@immd4.informatik.uni-erlangen.de)

Michael Schroeder (mlschroe@immd4.informatik.uni-erlangen.de)

Copyright (C) 1987 Oliver Laumann

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program (see the file COPYING); if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

CONTRIBUTORS

Ken Beal (kbeal@amber.ssd.csd.harris.com),

Rudolf Koenig (rfkoenig@immd4.informatik.uni-erlangen.de),

Toerless Eckert (eckert@immd4.informatik.uni-erlangen.de),
 Wayne Davison (davison@borland.com),
 Patrick Wolfe (pat@kai.com, kailand!pat),
 Bart Schaefer (schaefer@cse.ogi.edu),
 Nathan Glasser (nathan@brokaw.lcs.mit.edu),
 Larry W. Virden (lvirden@cas.org),
 Howard Chu (hyc@hamauma.jpl.nasa.gov),
 Tim MacKenzie (tym@dibbler.cs.monash.edu.au),
 Markku Jarvinen (mta@{cc,cs,ee}.tut.fi),
 Marc Boucher (marc@CAM.ORG),
 Doug Siebert (dsiebert@isca.uiowa.edu),
 Ken Stillson (stillson@tsfsrv.mitre.org),
 Ian Frechett (frechett@spot.Colorado.EDU),
 Brian Koehmstedt (bpk@gnu.ai.mit.edu),
 Don Smith (djs6015@ulb.isc.rit.edu),
 Frank van der Linden (vdlinden@fwi.uva.nl),
 Martin Schweikert (schweik@cpp.ob.open.de),
 David Vrona (dave@sashimi.lcu.com),
 E. Tye McQueen (tye%spillman.UUCP@uunet.uu.net),
 Matthew Green (mrg@eterna.com.au),
 Christopher Williams (cgw@pobox.com),
 Matt Mosley (mattm@access.digex.net),
 Gregory Neil Shapiro (gshapiro@wpi.WPI.EDU),
 Johannes Zellner (johannes@zellner.org),
 Pablo Averbuj (pablo@averbuj.com).

VERSION

This is version 4.0.2. Its roots are a merge of a custom version 2.3PR7 by Wayne Davison and several enhancements to Oliver Laumann's version 2.0. Note that all versions numbered 2.x are copyright by Oliver Laumann.

AVAILABILITY

The latest official release of *screen* available via anonymous ftp from gnudist.gnu.org, nic.funet.fi or any other *GNU* distribution site. The home site of *screen* is ftp.uni-erlangen.de, in the directory pub/utilities/screen. The subdirectory 'private' contains the latest beta testing release. If you want to help, send a note to screen@uni-erlangen.de.

BUGS

- 'dm' (delete mode) and 'xs' are not handled correctly (they are ignored). 'xn' is treated as a magic-margin indicator.
- *Screen* has no clue about double-high or double-wide characters. But this is the only area where *vttest* is allowed to fail.
- It is not possible to change the environment variable \$TERMCAP when reattaching under a different terminal type.
- The support of terminfo based systems is very limited. Adding extra capabilities to \$TERMCAP may not have any effects.
- *Screen* does not make use of hardware tabs.
- *Screen* must be installed as set-uid with owner root on most systems in order to be able to correctly change the owner of the tty device file for each window. Special permission may also be required to write the file "/etc/utmp".

- Entries in “/etc/utmp” are not removed when *screen* is killed with SIGKILL. This will cause some programs (like “w” or “rwho”) to advertise that a user is logged on who really isn’t.
- *Screen* may give a strange warning when your tty has no utmp entry.
- When the modem line was hung up, *screen* may not automatically detach (or quit) unless the device driver is configured to send a HANGUP signal. To detach a *screen* session use the -D or -d command line option.
- If a password is set, the command line options -d and -D still detach a session without asking.
- Both “breaktype” and “defbreaktype” change the break generating method used by all terminal devices. The first should change a window specific setting, where the latter should change only the default for new windows.
- When attaching to a multiuser session, the user’s .screenrc file is not sourced. Each user’s personal settings have to be included in the .screenrc file from which the session is booted, or have to be changed manually.
- A weird imagination is most useful to gain full advantage of all the features.
- Send bug-reports, fixes, enhancements, t-shirts, money, beer & pizza to **screen@uni-erlangen.de**.

NAME

scriptextract – extract scripts distributed with the edrc distribute command

SYNOPSIS

edrc/lib/edrc/scriptextract [**-h** | **-V**]

edrc/lib/edrc/scriptextract -m { **full** | **differential** } **-c** *config_file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

extract scripts distributed with the **distribute** EDRC command.

For a detail description of the distribution mechanism, see manpage of the **edrc(1m)** command.

OPTIONS

-h this usage message.

-V print scriptextract version.

-m *mode* install mode.

-c *config_file*
 configuration file.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---------------------------------------|
| 0 | no error. |
| 1 | command started using the wrong user. |
| 2 | operating system not supported. |
| 3 | command version printed. |

- 5** command aborted.
- 6** configuration file *config_file* not found.

FILES

edrc/var/repl/rec.config_file.full.cpio.gz

distribution file containing the received script files for a full distribution.

edrc/var/repl/rec.config_file.diff.cpio.gz

distribution file containing the received script files for a differential distribution.

edrc/var/lock/scriptextract.scriptextract.lock

lock file to ensure that only one instance of the **scriptextract** command is running.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc**(1m)

NOTES

-

BUGS

-

AUTHOR

scriptextract was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

scriptgrep – grep underlying recovery scripts for a pattern

SYNOPSIS

edrc/bin/scriptgrep [-l] *pattern*

AVAILABILITY

WA2L/edrc

DESCRIPTION

grep all recovery scripts for a pattern. Files in **.sav** directories and files not following the name convention ***.*** are excluded from the search.

OPTIONS

-l list only names of files containing matches.

pattern regular expression pattern to be searched for in underlying recovery scripts. See **regexintro(4)** for more information about regular expressions.

ENVIRONMENT

-

EXIT STATUS

4 no pattern defined, usage displayed.

X exit code of **grep(1)**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **filegrep(1)**, **grep(1)**, **regexintro(4)**

NOTES

-

BUGS

-

AUTHOR

scriptgrep was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

scriptheadersync – sync recovery script file header information to actual filename

SYNOPSIS

edrc/bin/scriptheadersync [**-h**]

scriptheadersync **-print** | **-execute**

AVAILABILITY

WA2L/edrc

DESCRIPTION

synchronize the header entries in recovery scripts with the actual filename of the script.

scriptheadersync starts in the current working directory and searches all underlying directories for environment files (*_env*) and recovery scripts matching the filename convention: *<menupoint>:<scriptname>*.

This command is useful if after cloning a menu and the scriptnames were renamed using the **lsmv**(1) or copied using the **lscp**(1) command.

OPTIONS

-h usage message.

-print print the intended changes to the script headers.

-execute apply the synchronization changes to the script headers.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 operating system not supported yet, see **osid**(3) if you get this error.

4 usage printed.

FILES

- .sav** directories where file saves are stored in recovery script trees. Those directories are excluded from header synchronization.
- _env** environment file.

EXAMPLES

-

SEE ALSO

edrcintro(1)

NOTES

-

BUGS

-

AUTHOR

scripthheadersync was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

scriptmenupath – return recovery script menu path

SYNOPSIS

edrc/lib/scriptmenupath [**-h**]

scriptmenupath [**-d**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return recovery script menu path with menu point when called from a recovery script or **_env** file in the form 'menu -> submenu -> subsubmenu -> menupoint'.

OPTIONS

-h usage message.

-d print the menu path without menu point in the form 'menu -> submenu -> subsubmenu'.

ENVIRONMENT

-

EXIT STATUS

0 always

1 the **scriptmenupath** command has been called from outside of an **edrc** session.

4 usage printed.

FILES

_env environment file where **scriptmenupath** is started. See **edrc(1m)** for more information.

<recovery script>

recovery script that is started, when a menupoint is called.

EXAMPLES

-

SEE ALSO

edrcintro(1), **scripttitle(3)**

NOTES

-

BUGS

-

AUTHOR

scriptmenupath was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

scriptrevision – revision (version) of a script- or text-file

SYNOPSIS

edrc/lib/scriptrevision *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

print the revision number of a (script)file following the [##] revision notation as also resolved by **revision(1)**.

OPTIONS

file file to resolve the version number from.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 given file does not exist or cannot be read.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), apprevision(3), revision(1)

NOTES

-

BUGS

-

AUTHOR

scriptrevision was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

scriptsequence – check if scripts are called in ascending order

SYNOPSIS

edrc/lib/scriptsequence [-h]

scriptsequence [-a inform | -a query | -a cancel | -a init]

AVAILABILITY

WA2L/edrc

DESCRIPTION

check if recovery scripts are called in ascending sequence. If not a message is printed or the script is aborted. **scriptsequence** must be called from a recovery script or **_env** file to give the user more awareness if the sequence is broken.

The **scriptsequence** command reads the **contrib.doc**(1m) tag **# R: text** to check if it is allowed to rerun a recovery script without a message/query.

If *text* in **# R: text** is set to **YES**, **JA**, **OUI** or **SI** and the related recovery script is started again, no message is printed or no query is performed.

OPTIONS

-a inform if the script is started out of ascending sequence or if a script is started again and no **# R: YES** is found in the recovery script documentation, print an information WARNING message, but start the recovery script nevertheless.

-a query if the script is started out of ascending sequence or if a script is started again and no **# R: YES** is found in the recovery script documentation, query the user if the script should be started or canceled.

-a cancel if the script is started out of ascending sequence or if a script is started again and no **# R: YES** is found in the recovery script documentation, cancel the recovery script start.

-a init force the **scriptsequence** command to re-read the script tree on next start.

Normally this is not necessary, but if recovery scripts were added/deleted the **scriptsequence** needs to re-read the script tree.

When the current **edrc** session is exited and a new session is started, it is also not needed to invoke **scripttree -a init**.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|--|
| 0 | no abnormality discovered. |
| 1 | scriptsequence is started from outside of a recovery script tree. |
| 2 | the script is started out of ascending sequence or the a script is started again and no # R: YES is found in the recovery script documentation and either -a cancel was invoked or the query to start/cancel the recovery script was answered to cancel. |
| 4 | usage displayed. |

FILES

_env environment file where **scriptsequence** is started. See **edrc(1m)** for more information.

<recovery script>

recovery script that is started, when a menupoint is called.

edrc/var/log/scriptsequence.log

log file of **scriptsequence**.

EXAMPLES

1. Usage example (query to abort or not):

```
scriptsequence || exit
```

2. Usage example (inform only):

```
scriptsequence -a inform || exit
```

SEE ALSO

edrcintro(1), **contrib.doc(1m)**, **contrib.edrc(1m)**, **contrib.scriptsequence(1m)**

NOTES

-

BUGS

-

AUTHOR

scriptsequence was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

scripttitle – print recovery script title and duration

SYNOPSIS

edrc/lib/scripttitle [**-h**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print recovery script title (menupoint, description and duration as defined in **doc** tag) when called from a recovery script or **_env** file to give the user more awareness which menu point is started.

OPTIONS

-h usage message.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|--|
| 0 | always |
| 1 | the scripttitle command has been called from outside of an edrc session. |
| 4 | usage printed. |

FILES

_env environment file where **scripttitle** is started. See **edrc(1m)** for more information.

<recovery script>

recovery script that is started, when a menupoint is called.

EXAMPLES

-

SEE ALSO

edrcintro(1), **contrib.doc(1m)**

NOTES

-

BUGS

-

AUTHOR

scripttitle was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

seconds – print seconds since Epoch for a given date and time

SYNOPSIS

edrc/lib/seconds [**-h**]

seconds *MM/DD/YYYY hh:mm:ss*

AVAILABILITY

WA2L/edrc

DESCRIPTION

Calculate seconds since the Epoch for a given date and time.

seconds is the inverse function of the **timer** command.

In conjunction with the **timer** command, the **seconds** command can be used to do some date and time calculation.

Use **timer** to evaluate the seconds since the Epoch for the current moment.

OPTIONS

-h usage message.

YYYY-MM-DD hh:mm:ss

date and time in format "Year-Month-Day Hour:Minute:Second".

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

- 1) print seconds since Epoch for Sun Mar 18 23:47:05 CET 2007

```
[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: seconds 2007-03-18 23:47:05

1174258025
```

- 2) print date for 1174258025 since the Epoch

```
[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: timer 1174258025

2007-03-18 23:47:05
```

- 3) print seconds since the Epoch of current point in time

```
[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: timer

1174258402
```

- 4) print the date and time when a 72 hour countdown that started at February 1st 2007 at 12:00:00 will end.

```
[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: begin='seconds 2007-02-01 12:00:00'

[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: end='expr $begin + 72 \* 60 \* 60'

[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: timer $end

2007-02-04 12:00:00
```

- 5) same as example 4, but in an one-line statement.

```
[ /home/mzv7t0 ]
[ mzv7t0@rh7mzv7t001 ][ksh]: \
timer `expr \`seconds 2007-02-01 12:00:00\` + 72 \* 60 \* 60`

2007-02-04 12:00:00
```

seconds(3)

Library Commands

seconds(3)

SEE ALSO

edrcintro(1), **date(1)**, **days(3)**, **is_weekend(3)**, **timer(1)**, **expr(1)**

NOTES

-

BUGS

-

AUTHOR

seconds was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sectioncat – cat a [SECTION] from a file/stream

SYNOPSIS

edrc/lib/sectioncat *SECTION* < *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

cat section ([SECTION]) of a text file.

OPTIONS

SECTION section (without braces) to be displayed from *file* or stream.

A the provided *SECTION* is converted to upper case.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **csvcat**(3)

NOTES

-

BUGS

-

AUTHOR

sectioncat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sed1line – useful one-line scripts for sed

SYNOPSIS

sed [*OPTION*]... *{script-only-if-no-other-script}* [*input-file*]...

AVAILABILITY

WA2L/edrc

DESCRIPTION**INTRODUCTION:**

```
-----
USEFUL ONE-LINE SCRIPTS FOR SED (Unix stream editor)          Dec. 29, 2005
Compiled by Eric Pement - pement[at]northpark[dot]edu          version 5.5
```

Latest version of this file (in English) is usually at:
<http://sed.sourceforge.net/sed1line.txt>
<http://www.pement.org/sed/sed1line.txt>

This file will also available in other languages:

```
Chinese      - http://sed.sourceforge.net/sed1line_zh-CN.html
Czech        - http://sed.sourceforge.net/sed1line_cz.html
Dutch        - http://sed.sourceforge.net/sed1line_nl.html
French       - http://sed.sourceforge.net/sed1line_fr.html
German       - http://sed.sourceforge.net/sed1line_de.html
Italian      - (pending)
Portuguese   - http://sed.sourceforge.net/sed1line_pt-BR.html
Spanish      - (pending)
```

FILE SPACING:

```
# double space a file
sed G
```

```
# double space a file which already has blank lines in it. Output file
# should contain no more than one blank line between lines of text.
sed '/^$/d;G'
```

```
# triple space a file
sed 'G;G'
```

```
# undo double-spacing (assumes even-numbered lines are always blank)
sed 'n;d'
```

```
# insert a blank line above every line which matches "regex"
sed '/regex/{x;p;x;}'
```

```
# insert a blank line below every line which matches "regex"
```

```
sed '/regex/G'
```

```
# insert a blank line above and below every line which matches "regex"
sed '/regex/{x;p;x;G;}'
```

NUMBERING:

```
# number each line of a file (simple left alignment). Using a tab (see
# note on '\t' at end of file) instead of space will preserve margins.
sed = filename | sed 'N;s/\n/\t/'
```

```
# number each line of a file (number on left, right-aligned)
sed = filename | sed 'N; s/^/      /; s/ *\(.\\{6,\\}\)\n/\1  /'
```

```
# number each line of file, but only print numbers if line is not blank
sed '/./=' filename | sed '/./N; s/\n/ /'
```

```
# count lines (emulates "wc -l")
sed -n '$='
```

TEXT CONVERSION AND SUBSTITUTION:

```
# IN UNIX ENVIRONMENT: convert DOS newlines (CR/LF) to Unix format.
sed 's/.$//' # assumes that all lines end with CR/LF
sed 's/^M$//' # in bash/tcsh, press Ctrl-V then Ctrl-M
sed 's/\x0D$//' # works on ssed, gsed 3.02.80 or higher
```

```
# IN UNIX ENVIRONMENT: convert Unix newlines (LF) to DOS format.
sed "s/$/'\`echo -e \\r\'/" # command line under ksh
sed 's/$/'"/`echo \\r`/'" # command line under bash
sed "s/$/'\`echo \\r\'/" # command line under zsh
sed 's/$/\r/' # gsed 3.02.80 or higher
```

```
# IN DOS ENVIRONMENT: convert Unix newlines (LF) to DOS format.
sed "s/$/" # method 1
sed -n p # method 2
```

```
# IN DOS ENVIRONMENT: convert DOS newlines (CR/LF) to Unix format.
# Can only be done with UnxUtils sed, version 4.0.7 or higher. The
# UnxUtils version can be identified by the custom "--text" switch
# which appears when you use the "--help" switch. Otherwise, changing
# DOS newlines to Unix newlines cannot be done with sed in a DOS
# environment. Use "tr" instead.
```

```
sed "s/\r/" infile >outfile # UnxUtils sed v4.0.7 or higher
tr -d \r <infile >outfile # GNU tr version 1.22 or higher
```

```
# delete leading whitespace (spaces, tabs) from front of each line
# aligns all text flush left
sed 's/^[ \t]*//'
```

```
# delete trailing whitespace (spaces, tabs) from end of each line
sed 's/[ \t]*$//'
```

```
# delete BOTH leading and trailing whitespace from each line
sed 's/^[ \t]*//;s/[ \t]*$//'
```

```

# insert 5 blank spaces at beginning of each line (make page offset)
sed 's/^/     /'

# align all text flush right on a 79-column width
sed -e :a -e 's/^\{1,78\}$ / & /;ta' # set at 78 plus 1 space

# center all text in the middle of 79-column width. In method 1,
# spaces at the beginning of the line are significant, and trailing
# spaces are appended at the end of the line. In method 2, spaces at
# the beginning of the line are discarded in centering the line, and
# no trailing spaces appear at the end of lines.
sed -e :a -e 's/^\{1,77\}$ / & /;ta' # method 1
sed -e :a -e 's/^\{1,77\}$ / & /;ta' -e 's/\( *\)\1/\1/' # method 2

# substitute (find and replace) "foo" with "bar" on each line
sed 's/foo/bar/' # replaces only 1st instance in a line
sed 's/foo/bar/4' # replaces only 4th instance in a line
sed 's/foo/bar/g' # replaces ALL instances in a line
sed 's/\(.*\)foo\(.*foo\)\/\1bar\2/' # replace the next-to-last case
sed 's/\(.*\)foo\/\1bar/' # replace only the last case

# substitute "foo" with "bar" ONLY for lines which contain "baz"
sed '/baz/s/foo/bar/g'

# substitute "foo" with "bar" EXCEPT for lines which contain "baz"
sed '/baz/!s/foo/bar/g'

# change "scarlet" or "ruby" or "puce" to "red"
sed 's/scarlet/red/g;s/ruby/red/g;s/puce/red/g' # most sed's
gsed 's/scarlet|ruby|puce/red/g' # GNU sed only

# reverse order of lines (emulates "tac")
# bug/feature in HHsed v1.5 causes blank lines to be deleted
sed '1!G;h;$!d' # method 1
sed -n '1!G;h;$p' # method 2

# reverse each character on the line (emulates "rev")
sed '/\n/!G;s/\(.\)\(.*\n\)\/&\2\1/;/D;s./'

# join pairs of lines side-by-side (like "paste")
sed '$!N;s/\n/ /'

# if a line ends with a backslash, append the next line to it
sed -e :a -e '/\$/N; s/\n\n//; ta'

# if a line begins with an equal sign, append it to the previous line
# and replace the "=" with a single space
sed -e :a -e '$!N;s/\n=/ /;ta' -e 'P;D'

# add commas to numeric strings, changing "1234567" to "1,234,567"
gsed ':a;s/\B[0-9]\{3\}\>/,/;ta' # GNU sed
sed -e :a -e 's/\([0-9]\{3\}\)/\1,2/;ta' # other sed's

# add commas to numbers with decimal points and minus signs (GNU sed)

```

```
gsed -r ':a;s/(\|[\^0-9.]) ([0-9]+) ([0-9]{3})/\1\2,\3/g;ta'
```

add a blank line every 5 lines (after lines 5, 10, 15, 20, etc.)

```
gsed '0~5G' # GNU sed only
sed 'n;n;n;n;G;' # other seds
```

SELECTIVE PRINTING OF CERTAIN LINES:

```
# print first 10 lines of file (emulates behavior of "head")
sed 10q

# print first line of file (emulates "head -1")
sed q

# print the last 10 lines of a file (emulates "tail")
sed -e :a -e '$q;N;11,$D;ba'
```

print the last 2 lines of a file (emulates "tail -2")

```
sed '$!N;$!D'
```

print the last line of a file (emulates "tail -1")

```
sed '$!d' # method 1
sed -n '$p' # method 2
```

print the next-to-the-last line of a file

```
sed -e '${h;d;}' -e x # for 1-line files, print blank line
sed -e '1{$q;}' -e '${h;d;}' -e x # for 1-line files, print the line
sed -e '1{$d;}' -e '${h;d;}' -e x # for 1-line files, print nothing
```

print only lines which match regular expression (emulates "grep")

```
sed -n '/regexp/p' # method 1
sed '/regexp/d' # method 2
```

print only lines which do NOT match regexp (emulates "grep -v")

```
sed -n '/regexp/!p' # method 1, corresponds to above
sed '/regexp/d' # method 2, simpler syntax
```

print the line immediately before a regexp, but not the line
containing the regexp

```
sed -n '/regexp/{g;1!p;};h'
```

print the line immediately after a regexp, but not the line
containing the regexp

```
sed -n '/regexp/{n;p;}'
```

print 1 line of context before and after regexp, with line number
indicating where the regexp occurred (similar to "grep -A1 -B1")

```
sed -n -e '/regexp/{=;x;1!p;g;$!N;p;D;}' -e h
```

grep for AAA and BBB and CCC (in any order)

```
sed '/AAA/!d; /BBB/!d; /CCC/!d'
```

grep for AAA and BBB and CCC (in that order)

```
sed '/AAA.*BBB.*CCC/!d'
```



```

# grep for AAA or BBB or CCC (emulates "egrep")
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d      # most sed's
gsed '/AAA\|BBB\|CCC/!d'                        # GNU sed only

# print paragraph if it contains AAA (blank lines separate paragraphs)
# HHsed v1.5 must insert a 'G;' after 'x;' in the next 3 scripts below
sed -e '/./{H;$!d;} -e 'x;/AAA/!d;'

# print paragraph if it contains AAA and BBB and CCC (in any order)
sed -e '/./{H;$!d;} -e 'x;/AAA/!d;/BBB/!d;/CCC/!d'

# print paragraph if it contains AAA or BBB or CCC
sed -e '/./{H;$!d;} -e 'x;/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
gsed '/./{H;$!d;};x;/AAA\|BBB\|CCC/b;d'          # GNU sed only

# print only lines of 65 characters or longer
sed -n '/^.\{65\}/p'

# print only lines of less than 65 characters
sed -n '/^.\{65\}/!p'          # method 1, corresponds to above
sed '/^.\{65\}/d'             # method 2, simpler syntax

# print section of file from regular expression to end of file
sed -n '/regexp/, $p'

# print section of file based on line numbers (lines 8-12, inclusive)
sed -n '8,12p'                # method 1
sed '8,12!d'                  # method 2

# print line number 52
sed -n '52p'                  # method 1
sed '52!d'                    # method 2
sed '52q;d'                   # method 3, efficient on large files

# beginning at line 3, print every 7th line
gsed -n '3~7p'                # GNU sed only
sed -n '3,$ {p;n;n;n;n;n;n;} ' # other sed's

# print section of file between two regular expressions (inclusive)
sed -n '/Iowa/,/Montana/p'     # case sensitive

```

SELECTIVE DELETION OF CERTAIN LINES:

```

# print all of file EXCEPT section between 2 regular expressions
sed '/Iowa/,/Montana/d'

# delete duplicate, consecutive lines from a file (emulates "uniq").
# First line in a set of duplicate lines is kept, rest are deleted.
sed '$!N; /\^(.*)\n\1$/!P; D'

# delete duplicate, nonconsecutive lines from a file. Beware not to
# overflow the buffer size of the hold space, or else use GNU sed.
sed -n 'G; s/\n/&&; /\^(.*)\n\1$/d; s/\n//; h; P'

# delete all lines except duplicate lines (emulates "uniq -d").

```

```

sed '$!N; s/^\(.*\)\\n\\1$/\\1/; t; D'

# delete the first 10 lines of a file
sed '1,10d'

# delete the last line of a file
sed '$d'

# delete the last 2 lines of a file
sed 'N;$!P;$!D;$d'

# delete the last 10 lines of a file
sed -e :a -e '$d;N;2,10ba' -e 'P;D'      # method 1
sed -n -e :a -e '1,10!{P;N;D;};N;ba'    # method 2

# delete every 8th line
gsed '0~8d'                             # GNU sed only
sed 'n;n;n;n;n;n;n;d;'                  # other seds

# delete lines matching pattern
sed '/pattern/d'

# delete ALL blank lines from a file (same as "grep '.' ")
sed '/^$/d'                             # method 1
sed '/./!d'                             # method 2

# delete all CONSECUTIVE blank lines from file except the first; also
# deletes all blank lines from top and end of file (emulates "cat -s")
sed '/./,/^$/!d'                        # method 1, allows 0 blanks at top, 1 at EOF
sed '/^$/N;/\\n$/D'                     # method 2, allows 1 blank at top, 0 at EOF

# delete all CONSECUTIVE blank lines from file except the first 2:
sed '/^$/N;/\\n$/N;///D'

# delete all leading blank lines at top of file
sed '/./,$!d'

# delete all trailing blank lines at end of file
sed -e :a -e '/^\\n*${$d;N;ba' -e '}'     # works on all seds
sed -e :a -e '/^\\n*$/N;/\\n$/ba'         # ditto, except for gsed 3.02.*

# delete the last line of each paragraph
sed -n '/^$/{p;h;};/./{x;/./p;}'

```

SPECIAL APPLICATIONS:

```

# remove nroff overstrikes (char, backspace) from man pages. The 'echo'
# command may need an -e switch if you use Unix System V or bash shell.
sed "s/.'echo \\b'//g"                  # double quotes required for Unix environment
sed 's/.'^H//g'                         # in bash/tcsh, press Ctrl-V and then Ctrl-H
sed 's/.'x08//g'                        # hex expression for sed 1.5, GNU sed, ssed

# get Usenet/e-mail message header
sed '/^$/q'                             # deletes everything after first blank line

```

```

# get Usenet/e-mail message body
sed '1,/^$/d'          # deletes everything up to first blank line

# get Subject header, but remove initial "Subject: " portion
sed '/^Subject: *!/d; s///;q'

# get return address header
sed '/^Reply-To:/q; /^From:/h; ./d;q;q'

# parse out the address proper. Pulls out the e-mail address by itself
# from the 1-line return address header (see preceding script)
sed 's/ *(.*)//; s/>.*//; s/.*[[:<]] *//'

# add a leading angle bracket and space to each line (quote a message)
sed 's/^/> //'

# delete leading angle bracket & space from each line (unquote a message)
sed 's/^> //'

# remove most HTML tags (accommodates multiple-line tags)
sed -e :a -e 's/<[^>]*>//g;/</N//ba'

# extract multi-part uuencoded binaries, removing extraneous header
# info, so that only the uuencoded portion remains. Files passed to
# sed must be passed in the proper order. Version 1 can be entered
# from the command line; version 2 can be made into an executable
# Unix shell script. (Modified from a script by Rahul Dhesi.)
sed '/^end/,/^begin/d' file1 file2 ... fileX | uuencode      # vers. 1
sed '/^end/,/^begin/d' "$@" | uuencode                      # vers. 2

# sort paragraphs of file alphabetically. Paragraphs are separated by blank
# lines. GNU sed uses \v for vertical tab, or any unique char will do.
sed '/./{H;d};x;s/\n/{NL}=/g' file | sort | sed '1s/{NL}=//;s/{NL}=/\n/g'
gsed '/./{H;d};x;y/\n/\v/' file | sort | sed '1s/\v//;y/\v/\n/'

# zip up each .TXT file individually, deleting the source file and
# setting the name of each .ZIP file to the basename of the .TXT file
# (under DOS: the "dir /b" switch returns bare filenames in all caps).
echo @echo off >zipup.bat
dir /b *.txt | sed "s/^\(.*\)\.TXT/pkzip -mo \1 \1.TXT/" >>zipup.bat

```

TYPICAL USE:

Sed takes one or more editing commands and applies all of them, in sequence, to each line of input. After all the commands have been applied to the first input line, that line is output and a second input line is taken for processing, and the cycle repeats. The preceding examples assume that input comes from the standard input device (i.e., the console, normally this will be piped input). One or more filenames can be appended to the command line if the input does not come from stdin. Output is sent to stdout (the screen). Thus:

```

cat filename | sed '10q'      # uses piped input
sed '10q' filename           # same effect, avoids a useless "cat"
sed '10q' filename > newfile # redirects output to disk

```

For additional syntax instructions, including the way to apply editing commands from a disk file instead of the command line, consult "sed & awk, 2nd Edition," by Dale Dougherty and Arnold Robbins (O'Reilly, 1997; <http://www.ora.com>), "UNIX Text Processing," by Dale Dougherty and Tim O'Reilly (Hayden Books, 1987) or the tutorials by Mike Arst distributed in U-SEDIT2.ZIP (many sites). To fully exploit the power of sed, one must understand "regular expressions." For this, see "Mastering Regular Expressions" by Jeffrey Friedl (O'Reilly, 1997). The manual ("man") pages on Unix systems may be helpful (try "man sed", "man regexp", or the subsection on regular expressions in "man ed"), but man pages are notoriously difficult. They are not written to teach sed use or regexps to first-time users, but as a reference text for those already acquainted with these tools.

QUOTING SYNTAX:

The preceding examples use single quotes ('...') instead of double quotes ("...") to enclose editing commands, since sed is typically used on a Unix platform. Single quotes prevent the Unix shell from interpreting the dollar sign (\$) and backquotes (`...`), which are expanded by the shell if they are enclosed in double quotes. Users of the "csh" shell and derivatives will also need to quote the exclamation mark (!) with the backslash (i.e., \!) to properly run the examples listed above, even within single quotes. Versions of sed written for DOS invariably require double quotes ("...") instead of single quotes to enclose editing commands.

USE OF '\t' IN SED SCRIPTS: For clarity in documentation, we have used the expression '\t' to indicate a tab character (0x09) in the scripts. However, most versions of sed do not recognize the '\t' abbreviation, so when typing these scripts from the command line, you should press the TAB key instead. '\t' is supported as a regular expression metacharacter in awk, perl, and HHsed, sedmod, and GNU sed v3.02.80.

VERSIONS OF SED:

Versions of sed do differ, and some slight syntax variation is to be expected. In particular, most do not support the use of labels (:name) or branch instructions (b,t) within editing commands, except at the end of those commands. We have used the syntax which will be portable to most users of sed, even though the popular GNU versions of sed allow a more succinct syntax. When the reader sees a fairly long command such as this:

```
sed -e '/AAA/b' -e '/BBB/b' -e '/CCC/b' -e d
```

it is heartening to know that GNU sed will let you reduce it to:

```
sed '/AAA/b;/BBB/b;/CCC/b;d'      # or even
sed '/AAA\|BBB\|CCC/b;d'
```

In addition, remember that while many versions of sed accept a command like "/one/ s/RE1/RE2/", some do NOT allow "/one/! s/RE1/RE2/", which contains space before the 's'. Omit the space when typing the command.

OPTIMIZING FOR SPEED:

If execution speed needs to be increased (due to large input files or slow processors or hard disks), substitution will be executed more quickly if the "find" expression is specified before giving the "s/.../.../" instruction. Thus:

```
sed 's/foo/bar/g' filename      # standard replace command
sed '/foo/ s/foo/bar/g' filename # executes more quickly
sed '/foo/ s//bar/g' filename   # shorthand sed syntax
```

On line selection or deletion in which you only need to output lines from the first part of the file, a "quit" command (q) in the script will drastically reduce processing time for large files. Thus:

```
sed -n '45,50p' filename      # print line nos. 45-50 of a file
sed -n '51q;45,50p' filename  # same, but executes much faster
```

CONTRIBUTIONS:

If you have any additional scripts to contribute or if you find errors in this document, please send e-mail to the compiler. Indicate the version of sed you used, the operating system it was compiled for, and the nature of the problem. To qualify as a one-liner, the command line must be 65 characters or less. Various scripts in this file have been written or contributed by:

```
Al Aab                # founder of "seders" list
Edgar Allen           # various
Yiorgos Adamopoulos  # various
Dale Dougherty       # author of "sed & awk"
Carlos Duarte         # author of "do it with sed"
Eric Pement          # author of this document
Ken Pizzini           # author of GNU sed v3.02
S.G. Ravenhall       # great de-html script
Greg Ubben           # many contributions & much help
```

OPTIONS

See: **sed**(1)

ENVIRONMENT

See: **sed**(1)

EXIT STATUS

See: **sed**(1)

FILES

See: **sed**(1)

EXAMPLES

-

SEE ALSO

edrcintro(1), **regexintro**(4), **sed**(1)

NOTES

The main part of this manpages is extracted from the documentation of **sed1line.txt** from <http://sed.sourceforge.net/sed1line.txt>.

BUGS

-

AUTHOR

sed1line.txt was developed by 'Eric Pemet' <pemente[at]northpark[dot]edu> (see: <http://sed.sourceforge.net/sed1line.txt>, <http://www.pement.org/sed/sed1line.txt>) and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports regarding to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

select_columns – select named columns from CSV stream

SYNOPSIS

edrc/lib/select_columns -h

select_columns "*field_separator*" "*column_list*"

AVAILABILITY

WA2L/edrc

DESCRIPTION

filter to select named columns that are separated by a field separator from **stdin** and print the selected columns to **stdout**. The first row of the data stream must contain the row header. Columns that do not exist, are printed as empty columns.

OPTIONS

-h usage message.

"*field_separator*"
field separator.

"*column_list*"
separated list of columns to be printed from the CSV stream from **stdin**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES**1) select certain columns form CSV data**

The following CSV data received via pipe

```
cat<<EOM | select_columns ";" "CNT;ZIP;CITY"
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

results in the output:

```
CNT;ZIP;CITY;
USA;93117;Goleta;
CH;8222;Beringen;
RSA;8222;Cape Town;
```

2) print selected columns of CSV data as formatted list

The columns CNT, ZIP and CITY of the CSV data are selected using the **select_columns** command and then formatted to a list using the **print_list(3)** command

```
cat<<EOM | select_columns ";" "CNT;ZIP;CITY" | print_list
ZIP;CITY;STATE;CNT;COUNTRY
93117;Goleta;CA;USA;United States of America
8222;Beringen;SH;CH;Switzerland
8005;Cape Town;WC;RSA;South Africa
EOM
```

what will result in the output:

```
CNT  ZIP    CITY
---  -
USA  93117   Goleta
CH   8222    Beringen
RSA  8005    Cape Town
(3)
```

SEE ALSO

edrcintro(1), **csv(3)**, **csvcat(3)**, **print_header(3)**, **print_index(3)**, **print_list(3)**

NOTES

The four commands **print_header(3)**, **print_list(3)**, **select_columns(3)** and **print_index(3)** provide the functionality to efficiently produce ASCII reports having an identical look.

select_columns(3)

Library Commands

select_columns(3)

BUGS

-

AUTHOR

select_columns was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`server_environment` – print server environment

SYNOPSIS

`edrc/lib/server_environment [-h | -l]`

server_environment

`server_environment [-d | -C | -R][-s servername]`

AVAILABILITY

WA2L/edrc

DESCRIPTION

Return a name for the environment of the server where running the command.

In the configuration file `edrc/etc/server_environment.cfg` it can be defined which server relates to what environment. The **server_environment** command can be used to develop scripts which act specific on different environments without hardcoding hostnames into the scripts.

When the server environment cannot be resolved, **unknown** is returned.

OPTIONS

return the name of the server environment where logged on.

-h usage message.

-d print the description of the server environment.

-C print customer name of the server environment.

-R print the whole info record in a **SERVER_ENVIRONMENT_<FIELD>='Value'** syntax that can be used to set variables in a script.

-l list all configured server environments.

-s *servername*

server to be resolved. For common usage it is not needed to specify the *servername* on the command line. This option can be used to verify configuration file definitions.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 1** information could not be resolved. In this case **unknown** is returned.
- 2** Operating system not supported. See **osid (3)** if you get this error.
- 4** usage printed.
- 6** configuration file **edrc/etc/server_environment.cfg** does not exist.

FILES

- edrc/etc/server_environment.cfg**
configuration file of **server_environment**. This file contains all server environment definitions.
- edrc/var/cache/server_environment/**
resolution cache of the **server_environment** command.
- edrc/var/cache/server_environment/meta**
meta data for the resolution cache. This file is used to check if the cache data is still valid.
- edrc/var/cache/server_environment/data**
this file contains the cache data, the resolved **CUSTOMER**, **NAME** and **DESCRIPTION** settings for the system where **server_environment** has been executed.
- edrc/var/samples/hostlist/**
configuration examples for **hostlist(3)** and **server_environment(3)**.

EXAMPLES

1) script cut out

In this script server environment dependent variable initialization is implemented using the **server_environment** command:

```
#!/bin/sh
:
:
:

case `server_environment` in
    TEST)
```

```

        Max_load=100
        Mail_to=fred.developer@acme.com
        ;;
    PREPRODUCTION)
        Max_load=200
        Mail_to=barney.verifier@acme.com
        ;;
    PRODUCTION)
        Max_load=250
        Mail_to=wilma.production@acme.com
        ;;
    unknown)
        echo "server environment unknown, aborting."
        exit 1
        ;;
esac

:
:
```

2) use of server_environment -R

In this example **server_environment -R** is used to efficiently query the whole information record and to print it.

```

#!/bin/sh
:
:

eval `server_environment -R`

cat <<EOM
Customer:    $SERVER_ENVIRONMENT_CUSTOMER
Environment: $SERVER_ENVIRONMENT_NAME
Description: $SERVER_ENVIRONMENT_DESCRIPTION
EOM

:
:
```

SEE ALSO

edrcintro(1), **server_environment.cfg(4)**, **hostlist(3)**, **edrc/var/samples/hostlist/**

NOTES

The **server_environment** command uses the **hostname(1)** command internally to resolve the hostname.

BUGS

-

AUTHOR

server_environment was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

server_environment.cfg – configuration file for server_environment

SYNOPSIS

edrc/etc/server_environment.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration for the **server_environment** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

NAME:Description:server_regex:Customer:

Rows starting with a # are considered as comments.

The definitions are processed in the sequence as listed in the configuration file. If a regular expression (*server_regex*) matches, the remaining definitions are not processed any more.

OPTIONS

NAME Name of the environment. It has been proven as a good practice to use all capital letters for the environment name, but this is not imperative. Characters other then "a-zA-Z0-9_.,=%+-" are replaced by an underscore (_) in the output.

Description
Free-form text to describe environment.

server_regex
Regular expression for server selection. If a servername passes the regular expression the related information (*NAME*, *Description*, *Customer*) is returned.

To test if a regular expression is working as expected, the command:

server_environment -s hostname

or

echo hostname | egrep "server_regex"

can be used.

Be aware, that the output of **hostname(1)** is checked against the *server_regex* when **server_environment** is called without the **-s** options.

Customer Name of the customer. This has to be a single word that can be used as file- and directory name, too. Characters other than "a-zA-Z0-9_.,=%+-" are replaced by an underscore (**_**) in the output.

EXAMPLES

1) normal example configuration file

This configuration file defines the six environments **PRODUCTION**, **PREPRODUCTION**, **TEST**, **MAINTENANCE**, **BACKUP_BE** and **BACKUP_BB** and is used in normal operations:

```
#
# etc/server_environment.cfg - cfg for server_environment in NORMAL OPE
# etc/server_environment.cfg.NORMAL_OPERATION - cfg for server_envirom
#
# [00] 24.07.2004 CWa Initial Version
#

#
# Format:
#   <NAME>:<Description>:<server_regex>:<Customer>:
#
PRODUCTION:GDC Bern, new PRODUCTION Environment:dcdbsi(2[0-1]|3[0-3]):A

PREPRODUCTION:GDC Balsberg, new PREPRODUCTION Environment:dcdbsi(50|6[1
TEST:GDC Balsberg, new TEST Environment:dcdbsi(6[3-4]|7[3-4]):ACME:
MAINTENANCE:GDC Balsberg, new MAINTENANCE Environment:dcdbsi(60|70):ACM

BACKUP_BE:GDC Bern, BACKUP Environment:dcdbsi(05|10|4[0-1]):ACME:
BACKUP_BB:GDC Balsberg, BACKUP Environment:dcdbsi(5[1-2]|8[0-1]):ACME:
```

2) disaster case configuration file

This configuration file defines the five environments **PRODUCTION**, **TEST**, **MAINTENANCE**, **BACKUP_BE** and **BACKUP_BB** and is used in a disaster case situation. This file replaces the file in example 1) during a disaster recovery where the **PREPRODUCTION** servers are used to host the **PRODUCTION** services.

Therefore the **PREPRODUCTION** definition disappeared and the **PRODUCTION** definition now contains the *server_regex* of the former **PREPRODUCTION**.

```
#
# etc/server_environment.cfg - cfg for server_environment in a DISASTER
# etc/server_environment.cfg.DISASTER_CASE - cfg for server_environment
```

```
#
# [00] 24.07.2004 CWa Initial Version
#

#
# Format:
#   <NAME>:<Description>:<server_regex>:<Customer>:
#
PRODUCTION:GDC Balsberg, DISASTER CENTER Environment:dcdbsi(50|6[1-2]|7

TEST:GDC Balsberg, new TEST Environment:dcdbsi(6[3-4]|7[3-4]):ACME:
MAINTENANCE:GDC Balsberg, new MAINTENANCE Environment:dcdbsi(60|70):ACM

BACKUP_BE:GDC Bern, BACKUP Environment:dcdbsi(05|10|4[0-1]):ACME:
BACKUP_BB:GDC Balsberg, BACKUP Environment:dcdbsi(5[1-2]|8[0-1]):ACME:
```

Due to the fact that **server_environment** reads the config file **edrc/etc/server_environment.cfg** it makes sense to create a symbolic link to the active configuration file if a disaster recovery situation has to be covered:

```
[ /opt/edrc/etc ]
[ root@dcdbsi30 ][*edrc*/bash]: ls -al server_environment.cfg*

lrwxrwxrwx    Jan 28 23:01 server_environment.cfg -> server_environment
-rw-r--r--    Feb 28 16:59 server_environment.cfg.DISASTER_CASE
-rw-r--r--    Apr  2 13:32 server_environment.cfg.NORMAL_OPERATION
```

See also **hostlist(3)** and **hostlist.cfg(4)** for more information about the special usage in a disaster case.

SEE ALSO

edrcintro(1), **server_environment(4)**, **hostlist(3)**, **hostlist.cfg(3)**, **hostlist.dat(4)**, **regexintro(3)**

NOTES

-

BUGS

-

AUTHOR

server_environment.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sh, jsh – the standard command interpreter

SYNOPSIS

sh [**-acefhikmnprstuvx**] [*arg*] ...

jsh [**-acefhikmnprstuvx**] [*arg*] ...

DESCRIPTION

Sh is a command programming language that executes commands read from a terminal or a file. See the *invocation* section for the meaning of arguments to the shell.

Commands

A *simple-command* is a sequence of non blank *words* separated by blanks (a blank is a **tab** or a **space**). The first word specifies the name of the command to be executed. Except as specified below the remaining words are passed as arguments to the invoked command. The command name is passed as argument 0 (see *exec(2)*). The *value* of a simple-command is its exit status if it terminates normally or 200+*status* if it terminates abnormally (see *signal(2)* for a list of status values).

A *pipeline* is a sequence of one or more *commands* separated by **|**. The standard output of each command but the last is connected by a *pipe(2)* to the standard input of the next command. Each command is run as a separate process; the shell waits for the last command to terminate. The *value* of a pipeline is the exit status of its last command.

A *list* is a sequence of one or more *pipelines* separated by **;**, **&**, **&&** or **||** and optionally terminated by **;** or **&**. **;** and **&** have equal precedence which is lower than that of **&&** and **||**, **&&** and **||** also have equal precedence. A semicolon causes sequential execution; an ampersand causes the preceding *pipeline* to be executed without waiting for it to finish. The symbol **&&** (**||**) causes the *list* following to be executed only if the preceding *pipeline* returns a zero (non zero) value. Newlines may appear in a *list*, instead of semicolons, to delimit commands.

A **#** at the beginning of a word starts a comment and causes the rest of the line to be ignored.

A *command* is either a simple-command or one of the following. The value returned by a command is that of the last simple-command executed in the command.

for name [in word ...] do list done

Each time a **for** command is executed *name* is set to the next word in the **for** word list. If **in word** ... is omitted then **in "\$@"** is assumed. Execution ends when there are no more words in the list.

case word in [pattern [pattern] ...) list ;;] ... esac

A **case** command executes the *list* associated with the first pattern that matches *word*. The form of the patterns is the same as that used for file name generation.

if list then [elif list then list] ... [else list] fi

The *list* following **if** is executed and if it returns zero the *list* following **then** is executed. Otherwise, the *list* following **elif** is executed and if its value is zero the *list* following **then** is executed. Failing that the **else list** is executed.

while list [do list] done

A **while** command repeatedly executes the **while list** and if its value is zero executes the **do list**; otherwise the loop terminates. The value returned by a **while** command is that of the last executed command in the **do list**. **until** may be used in place of **while** to negate the loop termination test.

(*list*) Execute *list* in a subshell.

{ *list*; } *list* is simply executed.

name() { *list*; }

Defines the shell function *name*. Each time when *name* is recognized as a command, *list* is executed, with the positional parameters *\$1*, *\$2*... set to the arguments of the command. After the function returns, the previous positional parameters are restored.

The following words are only recognized as the first word of a command and when not quoted.

if then else elif fi case in esac for while until do done { }

Command substitution

The standard output from a command enclosed in a pair of grave accents (``) may be used as part or all of a word; trailing newlines are removed.

Parameter substitution

The character **\$** is used to introduce substitutable parameters. Positional parameters may be assigned values by **set**. Variables may be set by writing

name=value [*name=value*] ...

\${parameter}

A *parameter* is a sequence of letters, digits or underscores (a *name*), a digit, or any of the characters * @ # ? - \$!. The value, if any, of the parameter is substituted. The braces are required only when *parameter* is followed by a letter, digit, or underscore that is not to be interpreted as part of its name. If *parameter* is a digit then it is a positional parameter. If *parameter* is * or @ then all the positional parameters, starting with **\$1**, are substituted separated by spaces. **\$0** is set from argument zero when the shell is invoked.

\${parameter:-word}

If *parameter* is set and not empty then substitute its value; otherwise substitute *word*.

\${parameter:=word}

If *parameter* is not set and not empty then set it to *word*; the value of the parameter is then substituted. Positional parameters may not be assigned to in this way.

\${parameter:?word}

If *parameter* is set and not empty then substitute its value; otherwise, print *word* and exit from the shell. If *word* is omitted then a standard message is printed.

\${parameter:+word}

If *parameter* is set and not empty then substitute *word*; otherwise substitute nothing.

If the **:** is omitted, the substitutions are only executed if the *parameter* is set, even if it is empty.

In the above *word* is not evaluated unless it is to be used as the substituted string. (So that, for example, `echo ${d-`pwd`}` will only execute `pwd` if `d` is unset.)

The following *parameters* are automatically set by the shell.

| | |
|-----------|--|
| # | The number of positional parameters in decimal. |
| - | Options supplied to the shell on invocation or by set . |
| ? | The value returned by the last executed command in decimal. |
| \$ | The process number of this shell. |
| ! | The process number of the last background command invoked. |

The following *parameters* are used by the shell:

CDPATH

The search path for the `cd` command (see above).

HOME The default argument (home directory) for the `cd` command.

OPTARG

The value of the last option argument processed by the `getopts` special command.

OPTIND

The index of the last option processed by the `getopts` special command.

PATH The search path for commands (see **execution**).

MAIL If this variable is set to the name of a mail file then the shell informs the user of the arrival of mail in the specified file.

MAILCHECK

If this variable is set, it is interpreted as a value in seconds to wait between checks for new mail. The default is 600 (10 minutes). If the value is zero, mail is checked before each prompt.

MAILPATH

A colon-separated list of files that are checked for new mail. MAIL is ignored if this variable is set.

PS1 Primary prompt string, by default '\$ '.

PS2 Secondary prompt string, by default '> '.

IFS Internal field separators, normally **space**, **tab**, and **newline**.

LANG, LC_ALL

See *locale(7)*.

LC_CTYPE

Affects the mapping of bytes to characters for file name generation, for the interpretation of '\', and for handling \$IFS.

SHACCT

If this variable is set in the initial environment passed to the shell and points to a file writable by the user, accounting statistics are written to it.

TIMEOUT

The shell exists when prompting for input if no command is entered for more than the given value in seconds. A value of zero means no timeout and is the default.

Blank interpretation

After parameter and command substitution, any results of substitution are scanned for internal field separator characters (those found in **\$IFS**) and split into distinct arguments where such characters are found. Explicit null arguments ("" or ``) are retained. Implicit null arguments (those resulting from *parameters* that have no values) are removed.

File name generation

Following substitution, each command word is scanned for the characters *, ? and [. If one of these characters appears then the word is regarded as a pattern. The word is replaced with alphabetically sorted file names that match the pattern. If no file name is found that matches the pattern then the word is left unchanged. The character . at the start of a file name or immediately following a /, and the character /, must be matched explicitly.

* Matches any string, including the null string.

? Matches any single character.

[...] Matches any one of the characters enclosed. A pair of characters separated by – matches any character lexically between the pair.

[!...] Matches any character except the enclosed ones.

Quoting

The following characters have a special meaning to the shell and cause termination of a word unless quoted.

;**&** (**)** | **^** **<** **>** **newline** **space** **tab**

A character may be *quoted* by preceding it with a \. **newline** is ignored. All characters enclosed between a pair of quote marks (`''), except a single quote, are quoted. Inside double quotes (" ") parameter and command substitution occurs and \ quotes the characters \ ` " and \$.

"\$*" is equivalent to "\$1 \$2 ..." whereas

"\$@" is equivalent to "\$1" "\$2"

Prompting

When used interactively, the shell prompts with the value of \$PS1 before reading a command. If at any time a newline is typed and further input is needed to complete a command then the secondary prompt (\$PS2) is issued.

Input and output

Before a command is executed its input and output may be redirected using a special notation interpreted by the shell. The following may appear anywhere in a simple-command or may precede or follow a *command* and are not passed on to the invoked command. Substitution occurs before *word* or *digit* is used.

< *word* Use file *word* as standard input (file descriptor 0).

> *word* Use file *word* as standard output (file descriptor 1). If the file does not exist then it is created; otherwise it is truncated to zero length.

>> *word*

Use file *word* as standard output. If the file exists then output is appended (by seeking to the end); otherwise the file is created.

<<[-] *word*

The shell input is read up to a line the same as *word*, or end of file. The resulting document becomes the standard input. If any character of *word* is quoted then no interpretation is placed upon the characters of the document; otherwise, parameter and command substitution occurs, **\newline** is ignored, and \ is used to quote the characters \\$ ` and the first character of *word*. The optional - causes leading tabulator character to be stripped from the resulting document; *word* may then also be prefixed by a tabulator.

< & *digit*

The standard input is duplicated from file descriptor *digit*; see *dup(2)*. Similarly for the standard output using > .

< & - The standard input is closed. Similarly for the standard output using > .

If one of the above is preceded by a digit then the file descriptor created is that specified by the digit (instead of the default 0 or 1). For example,

```
... 2>&1
```

creates file descriptor 2 to be a duplicate of file descriptor 1.

If a command is followed by & then the default standard input for the command is the empty file (/dev/null), unless job control is enabled. Otherwise, the environment for the execution of a command contains the file descriptors of the invoking shell as modified by input output specifications.

Environment

The environment is a list of name-value pairs that is passed to an executed program in the same way as a normal argument list; see *exec(2)* and *environ(5)*. The shell interacts with the environment in several ways. On invocation, the shell scans the environment and creates a *parameter* for each name found, giving it the corresponding value. Executed commands inherit the same environment. If the user modifies the values of these *parameters* or creates new ones, none of these affects the environment unless the *export* command is used to bind the shell's *parameter* to the environment. The environment seen by any executed command is thus composed of any unmodified name-value pairs originally inherited by the shell, plus any modifications or additions, all of which must be noted in *export* commands.

The environment for any *simple-command* may be augmented by prefixing it with one or more assignments to *parameters*. Thus these two lines are equivalent

```
TERM=450 cmd args
(export TERM; TERM=450; cmd args)
```

If the *-k* flag is set, *all* keyword arguments are placed in the environment, even if they occur after the command name. The following prints 'a=b c' and 'c':

```
echo a=b c
set -k
echo a=b c
```

Signals

The INTERRUPT and QUIT signals for an invoked command are ignored if the command is followed by & (unless job control is enabled); otherwise signals have the values inherited by the shell from its parent. (But see also *trap*.)

Execution

Each time a command is executed the above substitutions are carried out. The shell then first looks if a function with the command name was defined; if so, it is chosen for execution. Otherwise, except for the ‘special commands’ listed below a new process is created and an attempt is made to execute the command via an *exec*(2).

The shell parameter *\$PATH* defines the search path for the directory containing the command. Each alternative directory name is separated by a colon (:). The default path is ‘*/usr/5bin:/bin:/usr/bin:*’. If the command name contains a / then the search path is not used. Otherwise, each directory in the path is searched for an executable file. If the file has execute permission but is not an *a.out* file, it is assumed to be a file containing shell commands. A subshell (i.e., a separate process) is spawned to read it. A parenthesized command is also executed in a subshell.

Special commands

The following commands are executed in the shell process itself:

: No effect; the command does nothing.
.file Read and execute commands from *file* and return. The search path *\$PATH* is used to find the directory containing *file*.

break [*n*]

Exit from the enclosing *for* or *while* loop, if any. If *n* is specified then break *n* levels.

continue [*n*]

Resume the next iteration of the enclosing *for* or *while* loop. If *n* is specified then resume at the *n*-th enclosing loop.

cd [*arg*]

Change the current directory to *arg*. The shell parameter *\$HOME* is the default *arg*. If no directory *arg* is found and the *\$CDPATH* parameter contains a list of directories separated by colons, each of these directories is used as a prefix to *arg* in the given order, and the current directory is set to the first one that is found.

If there has still no suitable directory been found, an interactive shell will try to fix spelling errors and propose an alternative directory name:

```
$ cd /usr/lb
cd /usr/lib? y
ok
```

If the answer is ‘y’ or anything other than ‘n’, the shell will set the current directory to the one proposed.

echo [*arg ...*]

Each *arg* is printed to standard output; afterwards, a newline is printed. The following escapes sequences are recognized in *arg*:

\b Prints a backspace character.
\c Causes the command to return immediately. Any following characters are ignored, and the terminating newline is not printed.
\f Prints a formfeed character.
\n Prints a newline character.
\r Prints a carriage-return character.
\t Prints a tabulator character.
\v Prints a vertical tabulator character.
**** Prints a backslash character.
\0nnn Prints the character (byte) with octal value *nnn*.

If */usr/ucb* precedes */usr/5bin* or */usr/bin* in the current setting of the *\$PATH* variable and the first argument is **-n**, the terminating newline is not printed, and no escape sequences are recognized. If the *\$SYSV3* variable is set in the initial environment passed to the shell, the **-n** argument is also interpreted, but escape sequences are processed as usual.

eval [*arg* ...]

The arguments are read as input to the shell and the resulting command(s) executed.

exec [*arg* ...]

The command specified by the arguments is executed in place of this shell without creating a new process. Input output arguments may appear and if no other arguments are given cause the shell input output to be modified.

exit [*n*] Causes the shell to exit with the exit status specified by *n*. If *n* is omitted then the exit status is that of the last command executed. (An end of file will also exit from the shell.)

export [*name* ...]

The given names are marked for automatic export to the *environment* of subsequently-executed commands. If no arguments are given then a list of exportable names is printed.

getopts **optstring** **variable** [*arg* ...]

Retrieves options and option-arguments from *arg* (or the positional parameters) similar to *getopt*(3). *optstring* is a list of characters (bytes); each character represents an option letter. A character followed by **:** indicates that the option has an argument. Calling *getopts* repeatedly causes one option to be retrieved per call. The index of the current option is stored in the variable *OPTIND*; it is initialized to 1 when the shell starts. The option-argument, if any, is stored in the *OPTARG* variable. The option character is stored in the *variable* named. When the end of the options is reached, *getopts* returns with a non-zero value. A missing argument or an illegal option also causes a non-zero return value, and an error message is printed to standard error.

hash [*name* ...]

The shell maintains a hash table of the locations of external commands. If *name* arguments are given, each one is looked up and is inserted into the table if it is found. Otherwise, a list of the commands currently in the table is printed.

newgrp [*arg* ...]

Equivalent to 'exec newgrp *arg* ...'.

pwd Prints the name of the current directory.

read [**-r**] *name* ...

One line is read from the standard input; successive words of the input are assigned to the variables *name* in order, with leftover words to the last variable. The return code is 0 unless the end-of-file is encountered. Normally, backslashes in the line read escape the following character; this is inhibited if the **-r** option is given.

readonly [*name* ...]

The given names are marked readonly and the values of these names may not be changed by subsequent assignment. If no arguments are given then a list of all readonly names is printed.

return [*n*]

Return from a shell function to the execution level above. With the argument *n*, the special variable *\$?* is set to the given value.

set [**--aefhknptuvx**] [*arg* ...]

- No effect; useful if the first *arg* begins with **-**.
- a** Export any variables that are modified or created from now on.
- e** If non interactive then exit immediately if a command fails.
- f** File name generation is disabled.
- h** When a function is defined, look up all external commands it contains as described for the *hash* special command. Normally, these commands are looked up when they are executed.
- k** All keyword arguments are placed in the environment for a command, not just those that precede the command name.
- m** Enables job control (see below).
- n** Read commands but do not execute them.

- p** Makes the shell privileged. A privileged shell does not execute the system and user profiles; if a non-privileged shell (the default) has an effective user or group id different to its real user or group id or if it has an effective user or group id below 100, it resets its effective user or group id, respectively, to the corresponding real id at startup.
- t** Exit after reading and executing one command.
- u** Treat unset variables as an error when substituting.
- v** Print shell input lines as they are read.
- x** Print commands and their arguments as they are executed.
- Turn off the **-x** and **-v** options.

These flags can also be used upon invocation of the shell. The current set of flags may be found in **\$-**.

If **+** is used instead of **-**, the given flags are disabled.

Remaining arguments are positional parameters and are assigned, in order, to **\$1**, **\$2**, etc. If no arguments are given then the values of all names are printed.

shift [*n*]

The positional parameters from **\$2...** are renamed **\$1...** The *n* argument causes a shift by the given number, i.e. **\$n+1** is renamed to **\$1** and so forth.

times Print the accumulated user and system times for processes run from the shell.

test [*expr*]

test evaluates the expression *expr*, and if its value is true then returns zero exit status; otherwise, a non zero exit status is returned. *test* returns a non zero exit if there are no arguments.

The following primitives are used to construct *expr*:

- r** file true if the file exists and is readable.
- w** file true if the file exists and is writable.
- u** file true if the file exists and has the setuid bit set.
- g** file true if the file exists and has the setgid bit set.
- k** file true if the file exists and has the sticky bit set.
- f** file true if the file exists and is a regular file (or any file other than a directory if */usr/ucb* occurs early in the current **\$PATH** parameter).
- d** file true if the file exists and is a directory.
- h** file true if the file exists and is a symbolic link.
- L** file true if the file exists and is a symbolic link.
- p** file true if the file exists and is a named pipe.
- b** file true if the file exists and is a block device.
- c** file true if the file exists and is a character device.
- s** file true if the file exists and has a size greater than zero.
- t** [*fildev*] true if the open file whose file descriptor number is *fildev* (1 by default) is associated with a terminal device.
- z** *s1* true if the length of string *s1* is zero.
- n** *s1* true if the length of the string *s1* is nonzero.
- s1* = *s2* true if the strings *s1* and *s2* are equal.
- s1* != *s2* true if the strings *s1* and *s2* are not equal.
- s1* true if *s1* is not the null string.
- n1* **-eq** *n2* true if the integers *n1* and *n2* are algebraically equal. Any of the comparisons **-ne**, **-gt**, **-ge**, **-lt**, or **-le** may be used in place of **-eq**.

These primaries may be combined with the following operators:

- !** unary negation operator
- a** binary *and* operator
- o** binary *or* operator
- (*expr*) parentheses for grouping.

-a has higher precedence than **-o**. Notice that all the operators and flags are separate arguments

to *test*. Notice also that parentheses are meaningful as command separators and must be escaped.

trap [*arg*] [*n*|*name*] ...

Arg is a command to be read and executed when the shell receives signal(s) *n*. (Note that *arg* is scanned once when the trap is set and once when the trap is taken.) Trap commands are executed in order of signal number. If *arg* is absent then all trap(s) *n* are reset to their original values. If *arg* is the null string then this signal is ignored by the shell and by invoked commands. If *n* is 0 then the command *arg* is executed on exit from the shell, otherwise upon receipt of signal *n* as numbered in *signal*(2). *Trap* with no arguments prints a list of commands associated with each signal number. A symbolic *name* can be used instead of the *n* argument; it is formed by the signal name in the 'C' language minus the SIG prefix, e. g. TERM for SIGTERM. EXIT is the same as a '0' argument.

type *name* ...

For each *name*, prints if it would be executed as a shell function, as a special command, or as an external command. In the last case, the full path name to the command is also printed.

ulimit [-[HS][a|cd|fm|nstuv]]

ulimit [-[HS][c|d|f|m|n|s|t|u|v]] [*limit*]

Handles resource limits for the shell and processes created by it, as described in *getrlimit*(2). Without a *limit* argument, the current settings are printed; otherwise, a new limit is set. The following options are accepted:

- H Sets a hard limit. Only the super-user may raise a hard limit.
- S Sets a soft limit. A soft limit must not exceed the hard limit.
If neither -H or -S is given, the soft limit is printed, or both limits are set, respectively.
- a Chooses all limits described.
- c The maximum size of a core dump in 512-byte blocks.
- d The maximum size of the data segment in kbytes.
- f The maximum size of a file in 512-byte blocks. This is the default if no limit is explicitly selected.
- l The maximum size of locked memory in kbytes.
- m The maximum resident set size in kbytes.
- n The maximum number of open file descriptors.
- s The maximum size of the stack segment in kbytes.
- t The maximum processor time in seconds.
- u The maximum number of child processes.
- v The maximum address space size in kbytes.

umask [-S] [*nnn*]

The user file creation mask is set to the octal value *nnn* (see *umask*(2)). Symbolic modes as described in *chmod*(1) are also accepted. If *nnn* is omitted, the current value of the mask is printed. With the -S option, the current mask is printed as a symbolic string.

unset *variable* ...

Unsets each *variable* named.

wait [*n*]

Wait for the specified process and report its termination status. If *n* is not given then all currently active child processes are waited for. The return code from this command is that of the process waited for. If *n* does not refer to a child process of the shell, *wait* returns immediately with code 0.

Invocation

If the first character of argument zero is -, commands are read from */etc/profile* and *\$HOME/.profile*, if the respective file exists. Commands are then read as described below. The following flags are interpreted by the shell when it is invoked.

- c *string* If the -c flag is present then commands are read from *string*.
- s If the -s flag is present or if no arguments remain then commands are read from the standard input. Shell output is written to file descriptor 2.

- i** If the **-i** flag is present or if the shell input and output are attached to a terminal (as told by *isatty(3)*) then this shell is *interactive*. In this case the terminate signal SIGTERM (see *signal(2)*) is ignored (so that 'kill 0' does not kill an interactive shell) and the interrupt signal SIGINT is caught and ignored (so that **wait** is interruptable). In all cases SIGQUIT is ignored by the shell.

The remaining flags and arguments are described under the **set** command.

Job Control

When an interactive shell is invoked as **jsh**, job control is enabled. Job control allows to stop and resume processes, and to switch between foreground and background jobs. A job consists of the commands of a single pipeline. Each job is placed in a separate process group; a login shell and all jobs created by it form a session. Interrupt, quit, and other terminal control characters only affect the current foreground process group. The foreground job can be stopped pressing the suspend key, typically **^Z**; any job can be stopped by sending the STOP signal to it. Jobs are identified by *job ids* of the following form:

% or **% %** or **% +**

The current job.

% - The job that was previously the current job.

?string The only job whose name contains *string*.

%number

The job with the given number.

number

The job with process group id *number*.

string The only job for which *string* is a prefix of its name.

The following built-in commands are additionally available with job control:

bg [*jobid* ...]

Places each *jobid* in the background. The default job id is the current job.

fg [*jobid* ...]

Sequentially selects each *jobid* as the foreground job. The default job id is the current job.

jobs **[-p|-l]** [*jobid* ...] **[-x command [arguments ...]]**

Prints information about each *jobid*, or executes *command*.

-l Includes the process group id and the starting directory.

-p Includes the process group id.

-x command [arguments ...]

Executes *command* with *arguments*; each *argument* that forms a job id is replaced by the process group id of the respective job. It is an error if a given job does not exist.

kill **[[**-s** *signal* | **-signal**] *jobid* ... | **-l** [*status*]**

A special version of the *kill(1)* command that recognizes job ids in its arguments.

stop *jobid* ...

Stops the given jobs (i. e. sends a STOP signal to them).

suspend

Stops the shell itself. This is not allowed if the shell is a session leader.

wait [*jobid*]

The *wait* command (see above) recognizes job ids in its arguments.

FILES

/etc/profile

\$HOME/.profile

/tmp/sh*

/dev/null

SEE ALSO

env(1), exec(2), signal(2)

DIAGNOSTICS

Errors detected by the shell, such as syntax errors or fatal errors in special commands, cause the shell to return a non zero exit status. If the shell is being used non interactively then execution of the shell file is abandoned. Otherwise, the shell returns the exit status of the last command executed (see also *exit*).

NOTES

For historical reasons, ^ is a synonym for | as pipeline separator. Its use in new applications is discouraged.

If a command other than a simple-command (i. e. ‘for ...’, ‘case ...’ etc.) is redirected, it is executed in a sub-shell. If variable assignments must be visible in the parent shell after the input has been redirected, the *exec* special command can be used:

```
exec 5<&0 <input
while read line
do
    ...
    variable=value
    ...
done
exec <&5 5<&-
```

If parameters that have been inherited from the initial environment are modified, they must be explicitly exported to make the change visible to external commands, as described under ‘Environment’ above.

The IFS parameter is applied to any unquoted word. Thus

```
IFS=X
echoXfoo
```

executes the ‘echo’ command with the argument ‘foo’.

The command ‘set —’ without further arguments is a no-op. The *shift* special command can be used to delete all positional parameters.

There is only one namespace for both functions and parameters. A function definition will delete a parameter with the same name and vice-versa.

Parameter assignments that precede a special command affect the shell itself; parameter assignments that precede the call of a function are ignored.

NAME

shell – start the interactive shell

SYNOPSIS

edrc -> shell

edrc/bin/shell [**-h** | *host* | *user@host*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

There are two variants of the **shell** command. The first one is the **shell** command started within **edrc(1m)**, the second one is the **edrc/bin/shell** command line command.

EDRC command: edrc -> shell

start a shell in **edrc**. You should use this command to exit to the operating system rather than to do an other **telnet**, **rlogin** or **ssh** to the system.

The prompt is:

```
[ /opt/edrc/scripts/ACME-sat/file:SystemFiles ]
[ root@acme001 ][*edrc*/bash]:
```

When you use the **shell** command, the current working directory is the location in the menu you are currently in. Furthermore the shell settings are tuned and customized to the WA2L/edrc environment and all **\$EDRC_** environment variables are exported to the shell.

Call the **edrcenv(1)** command to display the official WA2L/edrc environment variables.

On UNIX the Korn-Shell (ksh), on Linux the Bourne-Again-Shell (bash) will be started. To see all additional commands and aliases you have available in the shell type **usage** within the started shell.

Command line command: edrc/bin/shell [*user@*] *host*]

the **shell** command provides a slightly reduced environment as the **shell** command started within **edrc**, but you can profit from the environment and all commands available in WA2L/edrc without the need to start **edrc(1m)**.

The prompt is:

```
[ /data/ACME/apps ]
[ root@acme001 ][*eshell*/bash]:
```

The differences to the **shell** start in **edrc** is, that the environment variables **\$EDRC_SESSION**, **\$EDRC_SCRIPTS_BASEDIR**, **\$EDRC_ENV**, **\$EDRC_CONFIGFILE**, **\$EDRC_RECOVERYTIME**, **\$EDRC_NLS_DATE_FORMAT**, **\$EDRC_NLS_LANG**, **\$NLS_DATE_FORMAT** and **\$NLS_LANG**

are not set.

Call the **edrcenv**(1) command to display the official WA2L/edrc environment variables.

When connecting to *host* the **shell** is started directly on the remote system if WA2L/edrc is installed. If WA2L/edrc is not installed a connection to the remote system without the **shell** enhancements is established.

COMMANDS

The following commands are only available when the **shell** is started (~built in commands / aliases). Those commands are intended to ease up the work when working interactively in the shell. Therefore this commands cannot be used within scripts. The commands described here are also listed in the **usage**(1) output, when **usage** is called within the **shell**.

handle working variables:

```

a [ something ]
    set working variable $a with the current working directory or something

b [ something ]
    set working variable $b with the current working directory or something

c [ something ]
    set working variable $c with the current working directory or something

d [ something ]
    set working variable $d with the current working directory or something

e [ something ]
    set working variable $e with the current working directory or something

f [ something ]
    set working variable $f with the current working directory or something

g [ something ]
    set working variable $g with the current working directory or something

abc
    print $a $b $c $d $e $f $g $p $TODAY

abc (a|b|c|d|e|f|g)
    load $a $b $c $d $e $f $g from other edrc/bin/shell or edrc/bin/eshell session

abcd
    print $a $b $c $d $e $f $g $p $TODAY

abcd (a|b|c|d|e|f|g)
    load $a $b $c $d $e $f $g from other edrc/bin/shell or edrc/bin/eshell session

```

\$PATH manipulation/querying:

```

addcwd    append current working directory to $PATH (export PATH=$PATH:'pwd')

cwdadd    add current working directory to $PATH (export PATH='pwd':$PATH)

ap        add ADDITIONAL_SHELL_PATH as defined in shell.cfg to $PATH (export
PATH=$PATH:$ADDITIONAL_SHELL_PATH)

path      print $PATH

```

change directories:

ret jump back to previous working directory

cda [*subdir*]
 cd \$a/subdir

cdb [*subdir*]
 cd \$b/subdir

cdc [*subdir*]
 cd \$c/subdir

cdd [*subdir*]
 cd \$d/subdir

cde [*subdir*]
 cd \$e/subdir

cdf [*subdir*]
 cd \$f/subdir

cdg [*subdir*]
 cd \$g/subdir

cdp [*subdir*]
 cd \$p/subdir

cdbin cd EDRC bin/

cdpbin cd EDRC pbin/

cdbkp cd EDRC BACKUP_DIR

cdbx [*subdir*]
 cd to user's DropBox directory (**USER_DROPBOX_DIR**=*dir* in **etc/shell.cfg**).

cdcnt cd EDRC contrib/edrc/

cddoc cd EDRC doc/

cdetc cd EDRC etc/

cdlib cd EDRC lib/

cdlog cd EDRC log/

cdman cd EDRC man/all/

cdscr cd EDRC scripts/

cdrel cd EDRC doc/.man/ which is the "release control" folder.

cdsrc cd EDRC src/

cdtmp cd EDRC \$EDRC_TMP_DIR. If the **\$EDRC_TMP_DIR** is not set, cd to **/tmp**.

cdsw cd EDRC software dir

cdvar cd EDRC var/

cdwww cd EDRC var/www/

cdrpt cd EDRC var/www/report/rpt/

cmc cd /etc/cmcluster/

debugging:

debug toggle script debug **\$DEBUG** between *True* and *False*

edrcdebug [*command...*]
toggle **\$EDRC_DEBUG**[*_COMMAND*] between *True* and *False*. It is possible to supply a space delimited list of commands.

session variables:

erase [*backspace*]
set tty erase character

term [*terminal*]
set/query **\$TERM**

termsize set **\$COLUMNS** and **\$LINES** based on the current terminal size.

tz [*timezone*]
set/query **\$TZ**

sid *ORACLE_SID*
set **\$ORACLE_SID** to *ORACLE_SID*

shd show **\$DISPLAY**

std *host* set **\$DISPLAY** to *host:0*

command history:

hist print shell history

r *hist_number* | *cmd*
re-execute *cmd* from history

histlist print a table of history files produced by **shell** when started outside of **shell** for the current user.

help and manual pages:

h [*cmd*] alias for **usage**(1)

help [*cmd*]
alias for **usage**(1)

man [*manpage*]
alias for **edrcman**(1)

manpath print **\$MANPATH**

list files and directories:

lc [*options*]
ls -aCF

lg [*options*]
ls -lagF

ll [*options*]
ls -laF

edit and view files:

vi [*options*] [*file ..*]
start **vi**(1) with prior initialization of the **\$COLUMNS** and **\$LINES** environment variables.

vim [*options*] [*file* ..]

start **vim**(1) with prior initialization of the **\$COLUMNS** and **\$LINES** environment variables.

view [*options*] [*file* ..]

start **view**(1) with prior initialization of the **\$COLUMNS** and **\$LINES** environment variables.

lgrep *pattern file*...

grep logfiles (having timestamps). The **lgrep** command is an alias to the **loggrep**(1) command.

manvi [*manpage*]

edit a manual page distributed in WA2L/edrc with **vi**(1). When editing a man page, first a backup copy in **.\$TODAY/** is created in the related **edrc/man/** subdirectory.

If no *manpage* is specified, the **edrcintro** man page is loaded into the editor.

The manual pages of the Perl modules bundled with WA2L/edrc cannot be edited with this command.

special commands:

exit exit the shell

ini *app* init environment for application *app*

lscolors toggle **ls** colors between *True* (= list files in color) and *False* (= do not list files in color).

grepcolors toggle **(e)grep** colors between *True* (= colorize results) and *False* (= do not colorize results).

pid select processes from the process command part of the process list. This is a convenience command to the **pslist**(1) command.

OPTIONS

Only command line variant.

-h usage message.

host system to connect to using SSH.

user@host using user on system to connect to.

ENVIRONMENT

For an explanation of all environment variables listed with **edrcenv**(1) see section **ENVIRONMENT** in **edrc(1m)**.

EXIT STATUS

| | |
|----------|---|
| 0 | no error. |
| 1 | shell is already started. |
| 2 | cannot find posix shell. |
| 4 | usage listed (only command line variant). |
| 5 | cannot connect to <i>host</i> . |

FILES

etc/kshrc main configuration file of **shell**. Do not edit this file, it might be patched on updates.

etc/shell.cfg
user configuration file of **shell**. This file can be adjusted.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc(1m)**, **bash(1)**, **ksh(1)**, **perl_modules(3)**, **shell.cfg(4)**

NOTES

This shell is not a programming shell, the only intention is to use it interactively.

BUGS

-

AUTHOR

shell was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2025 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

shell.cfg – user configuration file for shell

SYNOPSIS

edrc/etc/shell.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **shell**, **eshell** and **edrc shell** commands.

This file can be edited (the **etc/kshrc** file shouldn't.).

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **edrc**. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: LOG=var/log

Default: LOG=var/log

BACKUP_DIR

Backup directory of scriptextract. Here all script files are backedup, ahead of installing the new ones.

This setting is used when the **edbkp** command is used.

Example: BACKUP_DIR=var/backup

Default: BACKUP_DIR=var/backup

EDRC_DIST_USER

This is the user used to connect to remote hosts with **remote_shell**.

Example: EDRC_DIST_USER=\${EDRC_DIST_USER:=edrc}

Default: EDRC_DIST_USER

USER_DROPBOX_DIR

User's Dropbox directory.

This setting is used when the **cdbx** [*subdir*] command is used.

Example: USER_DROPBOX_DIR=~fred/var/dropbox

Default: USER_DROPBOX_DIR=

LC_TIME

Language specific settings.

Example: LC_TIME=C

Default: LC_TIME=

ADDITIONAL_SHELL_PATH

Additional \$PATH settings. This setting are applied to \$PATH when you invoke the **ap** command in the **shell**.

Example: ADDITIONAL_SHELL_PATH=~EDS/sh:~tools-eds/bin:~tools-eds/lib:/usr/local/bin:/usr/ccs/bin

Default: ADDITIONAL_SHELL_PATH=

SHELL_BANNER

Banner that is displayed when shell is executed outside of edrc.

Hint: When using commands to generate the banner, use the construct: "[\${EDRC_ENV}] || command" to speed up the shell execution when called within edrc.

Example: SHELL_BANNER="[\${EDRC_ENV}] || whereami 2>/dev/null"

Default: SHELL_BANNER=

README_DISPLAY

Display the **README** file (to **stderr**) that is present in a directory when changing the working directory to it using **cd**.

Example: README_DISPLAY=False

Default: README_DISPLAY=True

README_COLOR

Color (**RED**, **GREEN**, **BLUE**, **MAGENTA**, **CYAN**, **WHITE** or **YELLOW**) of the displayed **README** file.

Example: README_COLOR=RED

Default: README_COLOR=YELLOW

TPL_CFG

Additional config file for the **tpl**(1) command to add additional template locations beside the configurations in **etc/tpl.cfg** and **\$EDRC_SCRIPTS_BASEDIR/tpl.cfg**.

Example: TPL_CFG=/etc/system_templates.cfg

Default: TPL_CFG=\$HOME/etc/tpl.cfg

EDRC_SW_PATH

Path to WA2L/edrc software.

This path is resolved when invoking **cdsw** [*subdir*].

Example:

```
. 'aproot'/etc/edrcrevision.cfg
EDRC_SW_PATH="
'aproot'/var/sw/edrc-$VERSION.$PATCHLEVEL
/dat/sw/Generic/apps/edrc-patch
/dat/sw/apps/edrc-patch
/dat/sw/apps/edrc-$VERSION.$PATCHLEVEL
/software/tools/edrc-$VERSION.$PATCHLEVEL
/export/dat/sw/edrc-$VERSION.$PATCHLEVEL
/export/dat/sw
'aproot'/var/sw
"
```

Default: EDRC_SW_PATH=var/sw

SEE ALSO

edrcintro(1), **edrc**(1m), **eshell**(1), **shell**(1)

NOTES

-

BUGS

-

AUTHOR

shell.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

shellinaboxd – publish command line shell through AJAX interface

SYNOPSIS

shellinaboxd

```
[ -b | --background[=pidfile] ] [ --cert-fd=fd ] [ --css=filename ] [ --cgi[=portrange] ] [ -d |
--debug ] [ -f | --static-file=url:file ] [ -g | --group=gid ] [ -h | --help ] [ --linkify=[none|normal|aggressive] ] [ --localhost-only ] [ --no-beep ] [ -n | --numeric ] [ --pidfile=pidfile ] [ -p |
--port=port ] [ -s | --service=service ] [ --disable-ssl-menu ] [ -q | --quiet ] [ -u | --user=uid ]
[ --user-css=styles ] [ -v | --verbose ] [ --version ]
```

DESCRIPTION

The **shellinaboxd** daemon implements a webserver that listens on the specified *port*. The web server publishes one or more *services* that will be displayed in a VT100 emulator implemented as an AJAX web application. By default, the port is 4200 and the default service URL is *http://localhost:4200/*.

If no particular *service* was requested, the server launches **/bin/login** querying the user for their username and password. It then starts the user's default login shell.

Any modern JavaScript and CSS enabled browser will be able to access the published *service* without requiring additional plugins.

OPTIONS

The following command line parameters control the operation of the daemon:

-b | --background[=pidfile]

Launch **shellinaboxd** as a background daemon process. Optionally, write the process id to *pidfile*.

--css=filename

Sometimes, it is not necessary to replace the entire style sheet using the **--static-file** option. But instead a small incremental change should be made to the visual appearance of the terminal. The **--css** option provides a means to append additional style rules to the end of the default **styles.css** sheet. More than one **--css** option can be given on the same command line.

--cgi[=portrange]

Instead of running **shellinaboxd** as a permanent process, it can be demand-loaded as a CGI web server extension. When doing so, it will spawn a server that lives for the duration of the user's session. If an optional *portrange* of the form **MINPORT-MAXPORT** has been provided, the server limits itself to these port numbers. They should be configured to pass through the firewall.

The **--cgi** option is mutually exclusive with the **--background**, **--pidfile** and **--port** options.

In order to be useful as a CGI script, the **shellinaboxd** binary probably will have to be made **setuid-root**. This is currently a discouraged configuration. Use with care.

-d | --debug

Enables debugging mode, resulting in lots of log messages on *stderr*. This option is mutually exclusive with **--quiet** and **--verbose**.

-f | --static-file=url:file

The daemon serves various built-in resources from URLs underneath the *service* mount points. One or more **--static-file** options allow for overriding these resources with customized externally provided *files*. The *url* can either be an absolute or a relative path. In the former case, it overrides exactly one built-in resource for one specific *service*, whereas in the latter case it overrides resources for each defined *service*.

The following resources are available for customization:

| | |
|--------------------|---|
| beep.wav | audio sample that gets played whenever the terminal BEL is sounded. |
| favicon.ico | favicon image file that is displayed in the browser's navigation bar. |

ShellInABox.js JavaScript file implementing the AJAX terminal emulator.

styles.css CSS style file that controls the visual appearance of the terminal.

print-styles.css CSS style file that controls the visual appearance of printed pages when using the VT100 transparent printing feature.

It is not recommended to override the root HTML page for a particular *service*. Instead, move the service to an anonymous URL and serve a *static-file* that references the *service* in an `<iframe>`.

Instead of a *file*, it is possible to provide the name of a directory. This turns **shellinaboxd** into a simple web server that publishes all of the files in that particular directory. This option can be helpful when publishing a more complex root HTML page.

-g | --group=gid

When started as **root**, the server drops most privileges at start up. Unless overridden by the **--group** option, it switches to **nogroup**.

When already running as an unprivileged user, group changes are not possible.

If running with SSL/TLS support enabled, the certificates must be accessible to the unprivileged user and/or group that the daemon runs as.

-h | --help

Display a brief usage message showing the valid command line parameters.

--linkify=[none|normal|aggressive]

the daemon attempts to recognize URLs in the terminal output and makes them clickable. This is not necessarily a fool-proof process and both false negatives and false positives are possible. By default, only URLs starting with a well known protocol of **http://**, **https://**, **ftp://**, or **mailto:** are recognized. In **aggressive** mode, anything that looks like a hostname, URL or e-mail address is recognized, even if not preceded by a protocol.

--localhost-only

Normally, **shellinaboxd** listens on all available network interfaces. When operating behind a reverse-proxy that is not always desirable. This command line option tells the daemon to only listen on the loopback interface.

--no-beep

not only are audible signals undesired in some working environments, but browser support for media playback is often buggy, too. Setting this option suppresses all audio playback and enables the visual bell by default.

-n | --numeric

When running in **--verbose** mode, the daemon prints an *Apache*-style log file to *stderr*. By default, host names of peers get resolved before logging them. As DNS look-ups can be expensive, it is possible to request logging of numeric IP addresses, instead.

--pidfile=pidfile

The **shellinaboxd** daemon can be configured to store its process identifier in *pidfile*.

-p | --port=port

Unless overridden by this option, the web server listens on port 4200 for incoming HTTP and HTTPS requests.

shellinaboxd can distinguish between SSL/TLS requests and unencrypted requests. It also knows how to negotiate **Server Name Identification**, allowing the use of a single port for all types of requests even when virtual hosting.

-s | --service=service

One or more services can be registered on different URL paths:

SERVICE := <url-path> ':' *APPLICATION*

There is a pre-defined *application*, 'LOGIN', which causes the daemon to invoke **/bin/login** requesting the user's name and password, and starting his login shell. This is the default option for the **root** user, if no **--service** was defined. Starting **/bin/login** requires **root** privileges.

There is another pre-defined *application*, 'SSH'. Instead of invoking **/bin/login**, it calls **ssh**. This is the default option for unprivileged users, if no **--service** was defined. This operation is available to both privileged and regular users. If the optional *host* parameter is omitted, **shellinaboxd** connects to **localhost**.

Alternatively, an *application* can be specified by providing a *user* description, a working directory, and a command line:

```
APPLICATION := 'LOGIN' | 'SSH' [ ':' <host> ] | USER ':' CWD ':' CMD
```

```
USER := <username> ':' <groupname>
```

The working directory can either be given as an absolute path, or it can be the user's home directory:

```
CWD := 'HOME' : <dir>
```

The command that **shellinaboxd** executes can either be specified as the 'SHELL' keyword, denoting the user's default login shell, or an arbitrary command line:

```
CMD := 'SHELL' : <cmdline>
```

The <cmdline> supports expansion of variables of the form \${VAR}. Supported variables are:

\${columns} number of columns.

\${gid} numeric group id.

\${group} group name.

\${home} home directory.

\${lines} number of rows.

\${peer} name of remote peer.

\${uid} numeric user id.

\${url} the URL that serves the terminal session.

\${user} user name.

Other than the default environment variables of **\$TERM**, **\$COLUMNS** and **\$LINES**, services can have environment variables passed to them, by preceding the <cmdline> with space separated variable assignments of the form **KEY=VALUE**.

The <cmdline> supports single and double quotes, as well as backslashes for escaping characters in the familiar fashion.

Please note that when invoking **shellinaboxd** from a command line shell, additional quoting might be required to prevent the shell from expanding the variables prior to passing them to the daemon.

If no explicit **--service** has been requested, **shellinaboxd** defaults to attaching the default service to the root directory of the web server. For **root**, this is **/bin/login**, and for unprivileged users, this is **ssh localhost**. This is equivalent to saying **--service=/:LOGIN**, or **--service=/:SSH**, respectively.

--disable-ssl-menu

If the user should not be able to switch between HTTP and HTTPS modes, this choice can be removed from the context menu. The user can still make this choice by directly going to the appropriate URL.

-q | --quiet

Suppresses all messages to *stderr*. This option is mutually exclusive with **--debug** and **--verbose**.

-u | --user=uid

If started as **root**, the server drops privileges by changing to **nobody**, unless the *uid* has been overridden by this option.

For more details, refer to the description of the **--group** option.

--user-css=styles

The visual appearance of the terminal emulator can be customized through user-selectable style sheets. These style sheets will show up as options in the right-click context menu of the terminal emulator.

Styles sheet make up either independently selectable on/off options, or multiple style sheets can be grouped together. When forming a group, only one member of the group can be active at any given time. This is used for multiple-choice options.

Multiple independent groups are separated by semicolons:

```
STYLES := GROUP { ',' GROUP }*
```

The members of a group are separated by commas:

```
GROUP := OPTION { ',' OPTION }*
```

Groups with exactly one member are used for options that can be independently turned on and off.

Options include a human readable label that will be shown in the context menu, followed by the name of the CSS file. They also must include an indicator showing whether the option should initially be turned on or turned off. Within a group, exactly one option should be turned on:

```
OPTION := <label> ':' [ '-' | '+' ] <css-file>
```

The user's selection of options will be persisted in a cookie. This means, the default settings of options as passed on the command line only takes effect the very first time the user visits the terminal emulator in his browser. On all subsequent visits, the user's preferences take precedence.

-v | --verbose

Enables logging of *Apache*-style log file to *stderr*. This option is mutually exclusive with **--debug** and **--quiet**.

--version

Prints the version number of the binary and exits.

CONFIGURATION

There are no configuration files or permanent settings for **shellinaboxd**.

A small number of run-time configuration options are available from a context menu that becomes available when clicking the right mouse button. These options get persisted in a browser cookie.

Many sites already have a web server running and would like to integrate **shellinaboxd** into their existing site. This is most commonly done by means of a reverse-proxy entry for the main web server. For *Apache* this would require adding an option such as:

```
<Location /shell>
  ProxyPass http://localhost:4200/
  Order allow,deny
  Allow from all
</Location>
```

If you are using a different web server, refer to that server's documentation on how to configure reverse

proxy operations.

When using a reverse proxy, the **--localhost-only** option would normally be enabled as well. In addition, the **--disable-ssl** might also be considered depending on the exact configuration details of the reverse proxy.

EXAMPLES

shellinaboxd

Attaches a web-enabled login shell to *https://localhost:4200/*. If the user connected without SSL, the session will automatically be promoted. Unless SSL certificates can be found in the current directory, the daemon will automatically generate suitable self-signed certificates. If the command was invoked by a non-**root** user, the daemon uses **ssh** instead of **/bin/login** for the session.

shellinaboxd -t

Attaches a web-enabled login shell to *http://localhost:4200/* with SSL/TLS support disabled.

shellinaboxd -t -f beep.wav:/dev/null

Runs all services with the audible-bell permanently disabled.

shellinaboxd -s /:SSH:example.org

The terminal connects to a **ssh** session on *example.org*.

shellinaboxd -t -s /:AUTH:HOME:/bin/bash

Interactively request the user's name and password prior to launching a Bourne shell. This command can be run by unprivileged users. But if doing so, it only allows this particular user to log in.

shellinaboxd -c certificates -u shellinabox -g shellinabox

If the **certificates** directory exists and is writable by the **shellinabox** user and group, self-signed SSL certificates will be generated in this directory. This might require creating an appropriately named user first. Running this command as **root** allows any user on the system to log in at *http://localhost:4200/*. Sessions will automatically be promoted to SSL/TLS.

shellinaboxd -t -s /:LOGIN -s /who:nobody:nogroup:/w

In addition to the login shell at *http://localhost:4200*, show a list of currently logged in users when accessing *http://localhost:4200/who*. This command must be run as **root** in order to be able to change to **nobody:nogroup** as requested by the service description.

shellinaboxd -t -s '/:root:root:/wy60 -c /bin/login'

Instead of the standard **ANSI/VT100** terminal, publish a **Wyse 60™** terminal. Again, this command should be run as **root**.

shellinaboxd --css white-on-black.css

Loads the **white-on-black.css** style sheet from the current directory and appends it to the built-in **styles.css** sheet. This causes the terminal to always render white text on a black background.

shellinaboxd --user-css Normal:+black-on-white.css,Reverse:-white-on-black.css

Allow the user to select whether they want text to be rendered normally or in reverse video. This command line option adds a new entry to the right-click context menu.

DIAGNOSTICS

The daemon returns a non-zero exit code in case of failure. With the exception of a small number of common error cases that are handled explicitly, most errors result in printing a *"Check failed"* message. This does not typically indicate a bug in the program but is instead its normal way of reporting errors.

Common failure conditions are reusing a port that is already in use, lack of sufficient privileges to run a service, failure to find SSL/TLS certificates, and failure to write newly generated certificates to the certification directory.

SEE ALSO

chmod(1), last(1), login(1), sh(1), shells(5), openssl(1SSL), w(1), wy60(1), xterm(1).

SECURITY

The daemon uses privilege separation techniques to allow it to drop privileges early. It is aware of `setuid` flags and restricts some operations when launched as a `setuid` application.

Despite these safety features, a bug could conceivably lead to a determined attacker gaining elevated privileges. It is therefore strongly discouraged to set the `setuid` flag on the binary.

The expected deployment would be from a system `rc` script launched by `/sbin/init`. For extra security, the `--group` and `--user` options should be used to change to a dedicated user.

AUTHOR

Copyright (C) 2008-2010 by Markus Gutschke <markus@shellinabox.com>.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

In addition to these license terms, the author grants the following additional rights:

If you modify this program, or any covered work, by linking or combining it with the OpenSSL project's OpenSSL library (or a modified version of that library), containing parts covered by the terms of the OpenSSL or SSLeay licenses, the author grants you additional permission to convey the resulting work. Corresponding Source for a non-source form of such a combination shall include the source code for the parts of OpenSSL used as well as that of the covered work.

You may at your option choose to remove this additional permission from the work, or from any part of it.

If you would like to negotiate different licensing terms that are compatible for integration with other projects, please contact the author.

BUGS

Due to browser limitations, some features might not be available to users of all browsers.

Konqueror does not allow for reliable interception of `CTRL` keys. If you press a key together with the `CTRL` modifier, it continues performing the browser's predefined behavior for this particular key combination. In most cases, it also fails to report the correct key to the terminal emulator. As a work-around, pressing both the `CTRL` and the `WINDOWS` key sometimes works.

Some browsers, most notably IE on Windows, disallow interception of `ALT` keys and always interpret these keys as menu accelerators. As a work-around, many UNIX applications allow pressing `ESC`, instead of `ALT`.

When using non-US keyboard layouts, some browser do not allow for reliably determining shifted `ALT` keys. Please report these cases if they turn out to be a problem, as work-arounds might be possible.

Access to the native clipboard is typically not possible. Instead, an internal clipboard accessible from the right-button context menu is used for all but IE.

Some browsers restrict the number of concurrent connections to a server. This restricts how many AJAX terminals can be opened simultaneously. If this becomes a problem, users can typically reconfigure their browsers to raise the limit.

There have been reports of the VLC plugin on Linux/x86_64 crashing Firefox when the browser page gets reloaded. Setting the `--no-beep` option eliminates all references to VLC and thus appears to work around this crash.

NAME

shlib – print shared library path environment

SYNOPSIS

edrc/lib/shlib [-n]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print the environment variables needed by the dynamic linker **ld**(1) to locate the shared libraries. If a script need to set this environment variables (**\$LD_LIBRARY_PATH** and **\$SHLIB_PATH**), use

```
eval `shlib`
```

instead of setting them by your own. This ensures, that all scripts will continue to run, even when the directory structure in WA2L/edrc changes.

OPTIONS

-n no **export** *VARIABLE* ... output.

ENVIRONMENT**\$LD_LIBRARY_PATH**

path of additional shared libraries loaded by the **ld** dynamic linker. This variable will be expanded with the shared library locations for a certain Os-Id distributed with WA2L/edrc when executing the **shlib** command.

\$SHLIB_PATH

this variable is expanded as the **\$LD_LIBRARY_PATH** is done. This variable has basically the same function as the **\$LD_LIBRARY_PATH** and is mostly active on HP-UX systems.

EXIT STATUS

0 always

FILES

[**edrc/lib/<Os-Id>/<GLIBC>/libs/**]**edrc/lib/<Os-Id>/libs/**

location of operating system dependent shared libraries. This directory is appended to the **\$LD_LIBRARY_PATH** and the **\$SHLIB_PATH** environment variables when executing **shlib**.

/usr/ccs/lib:/lib:/usr/lib:/usr/openwin/lib:/usr/local/lib
standard library path for Solaris.

\$LD_LIBRARY_PATH:/etc:/lib:/lib/tls:/usr/lib:/usr/X11R6/lib:/usr/local/lib
standard library path for Linux.

\$LD_LIBRARY_PATH \$SHLIB_PATH
standard library path for HP-UX.

\$LD_LIBRARY_PATH:/lib:/usr/lib:/usr/local/lib \$SHLIB_PATH:/lib:/usr/lib:/usr/local/lib
standard library path for other operating systems.

EXAMPLES

-

SEE ALSO

edrcintro(1), **glibc.version(3)**, **ld(1)**, **osid(3)**, **os_wrapper(1)**

NOTES

-

BUGS

-

AUTHOR

shlib was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sortc – sort data using the "C" locale

SYNOPSIS

edrc/lib/sortc [*sort-options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

sorting data in the "C" locale.

Internally the **sort**(1) command is called with the environment variable **LC_ALL=C** set.

OPTIONS

sort-options

all supported options of the **sort**(1) command.

ENVIRONMENT

-

EXIT STATUS

x exit code of the **sort**(1) command.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **locale**(1), **sort**(1), **sortv**(3)

NOTES

-

BUGS

-

AUTHOR

sortc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sortv – portable version sort (sort -V)

SYNOPSIS

edrc/lib/sortv [*sort-options*] < *data*

AVAILABILITY

WA2L/edrc

DESCRIPTION

Provide a version sort functionality as provided by the **sort -V** command to operating systems, where **sort(1)** lacks this functionality.

On operating systems providing the **sort -V** functionality, the **sort -V** command is used internally.

OPTIONS

sort-options

all supported options of the **sort(1)** command.

ENVIRONMENT

see **sort(1)**.

EXIT STATUS

x exit code of the **sort(1)** command.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **locale(1)**, **sort(1)**, **sortc(1)**

NOTES

-

BUGS

-

AUTHOR

sortv was developed by Christian Walther. Send suggestions and bug reports to wa21@users.sourceforge.net

.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sparse – filter to identify sparse files

SYNOPSIS

edrc/lib/sparse [**-h**]

sparse [**-v**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

filter to identify sparse files.

The file list received thru **stdin** is searched for sparse files and the file names of all identified sparse files, or all non-sparse file names (**-v**), are printed to **stdout**. This enables to use the output directly for further processing thru a pipe for instance.

OPTIONS

-h print usage message.

-v print all non-sparse files. This is the inverse output, as if no option is specified.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 could not stat all files in provided file list.

4 usage message printed.

FILES

-

EXAMPLES**1) Find sparse files in the current directory**

```
find . -print | sparse
```

2) Copy all sparse files

```
find . -print | sparse | cpio -pdvm --sparse /destination
```

3) Copy all non-sparse files

```
find . -print | sparse -v | cpio -pdvm /destination
```

SEE ALSO

edrcintro(1) **cpio(1)**, **directories(3)**, **find(1)**, **stat(3)**, http://en.wikipedia.org/wiki/Sparse_file

NOTES

-

BUGS

-

AUTHOR

sparse was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

splitvt - run two shells in a split window

SYNOPSIS

splitvt [**options**] [**shell**]

OPTIONS

-upper command

Runs 'command' in the upper window

-lower command

Runs 'command' in the lower window

-s numlines

Sets 'numlines' to the number of lines in the top window. This number will be modified if the screen isn't big enough to handle the full size.

-t title Sets an xterm title bar to 'title'

-login Runs the programs under each window as though they were login shells. This is useful to have csh source it's .login file, bash source .profile, etc.

-nologin

Overrides any "set login on" statement in the startup file.

-rcfile file

Loads 'file' as the startup file instead of ~/.splitvtrc.

-norc Suppresses loading your ~/.splitvtrc

-v Print out the version number of splitvt

DESCRIPTION

This program splits the screen into two windows, one above the other, and runs a shell in each one. The default shell is taken from the SHELL environment variable, or /bin/csh if SHELL isn't in the environment, but another shell can be specified on the command line.

The top window is the default current window, but you can change to the other by pressing ^W. Pressing ^V will quote the next character to the current shell. Pressing ^O will put splitvt into command mode.

These special characters (command mode, switch window, and quote next character) can be modified from within the .splitvtrc file.

When in command mode, splitvt will read one non-numeric character, perform the requested command, and then revert to normal operations. Numeric characters are interpreted as a parameter for the specified command. The currently supported commands are:

'h' Print a help window for the escape commands

'-' Shrink the current window

'+' Expand the current window

'c' Select a block of text in the current window

'p' Paste the selected text to the current window

'k' Kill the current window (hangup signal)

'x' Lock the screen, after prompting for password

'r' Repaint the screen (clearing screen garbage)

'q' Quickly quit splitvt, killing the running shells

More functions may be added in the future.

If the -s command line option is used, the upper window will be bound to the user defined number of lines even if the screen is resized. Otherwise, the screen will be split into two equal parts based on the new size of the screen.

The environment variable SPLITVT is set in the shells forked by splitvt. In the shell running in the upper window, this variable is set to the value "upper", while the shell running in the lower window has this variable set to the value "lower". This is useful in shell scripts to provide different behavior depending on which window the script is running in. An example of this is shown in the file "menu" in the examples directory in the splitvt distribution.

When splitvt starts up, it looks in your home directory for a file named *.splitvtrc*. You can set a number of parameters from within this file, including all of the special characters, the default number of lines in the upper window, whether or not to run the commands as login shells, and even default commands to run in each window.

Here is an example of a *.splitvtrc* file:

```
# This is an example .splitvtrc file.
set command_char      ^O
set quote_char        ^V
set switch_char        ^W
set upper_lines        12
run -upper             /bin/sh
run -lower             top
set login              on
# This next line would override the above run statements
#run /bin/tcsh
```

If programs like vi are doing strange things to the window while in splitvt, you probably need to set the LINES environment variable to a correct value, or set the terminal's window sizing correctly. This should be done automatically, but may need to be corrected.

If you are running under an xterm window, the title bar will probably be updated to show the version of splitvt that you are running. Under a true xterm window, you can also switch which half of the window you are typing into by clicking the mouse in the half of the window in which you want to type. You can also drag and drop the separator bar by pressing down on the mouse button over the separator bar, holding it down, moving the mouse to where you want the bar to end up, and then releasing the button.

splitvt will attempt to erase the current utmp entry, and replace it with entries for the two windows. This allows you to use programs such as 'talk' within the splitvt windows. If you do not have write permission to the /etc/utmp file, you will not be able to modify the utmp entries.

splitvt can be made set-uid root. splitvt will reset its user id to that of the person running it, just before it exec()'s the shell under the window. The splitvt process remains with root permissions, and will change ownership of the pseudo terminals to that of the person running splitvt, and then reset it to root when the window is closed.

SPLITVT IS NOT GUARANTEED TO BE A SAFE SET-UID PROGRAM!

I have done all I know to keep splitvt a safely usable set-uid program, but I do not know everything, and am not responsible for any security weaknesses splitvt might possess.

BUGS

When used with a certain public domain version of ksh, you have to manually kill -9 on the shell processes after you quit splitvt. This has to do with a bug in that version of ksh (Version 06/03/86a and possibly others).

If splitvt is compiled with xterm support, when run under an xterm, the cut and paste feature is semi-disabled. The title bar may also be reset to the string "xterm" after splitvt quits. See the comments at the top of splitvt.c for how to disable xterm support.

Ansi colors are not repainted on screen refresh.

CAVEATS

There may be conflicts between splitvt's characters and other programs. You can either change splitvt's default characters with command line options, or you can type Ctrl-V (the quote character) within splitvt, and the next character typed will be sent to the current window, no matter what it is.

splitvt can only be used by terminals using vt100 emulation, due to the use of specific vt100 escape codes. (vt300, xterm, Linux console, etc, emulation also works)

When running in an xterm, if you press the escape key, you will have to type another character in order for the escape to be seen by the running program.

AUTHOR

Sam Lantinga slouken@cs.ucdavis.edu

NAME

sqlite – command-line access program for SQLite databases

SYNOPSIS

edrc/bin/sqlite [**-help** | **-version**]

sqlite [**-init** *filename*] [**-echo**] [**-[no]header**] [**-bail**] [**-interactive**] [**-batch**] [**-column**] [**-csv**] [**-html**] [**-line**] [**-list**] [**-separator** 'x'] [**-nullvalue** 'text'] [*FILENAME* [*SQL*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

SQLite is the most widely deployed SQL database engine in the world. It is used in countless desktop computer applications as well as consumer electronic devices including cellphones, PDAs, and MP3 players. The source code for SQLite is in the public domain.

The SQLite library includes a simple command-line utility named **sqlite3** that allows the user to manually enter and execute SQL commands against an SQLite database.

In the WA2L/edrc package the command is provided under the name **sqlite**. Therefore whenever the command **sqlite3** is referenced in the HTML documentation, replace it with **sqlite**.

This man page provides a brief introduction to the options of **sqlite** and provides the link to the HTML documentation.

OPTIONS

-help usage message.

-echo print commands before execution.

-[no]header turn headers on or off.

-bail stop after hitting an error.

-interactive force interactive I/O.

-batch force batch I/O.

-column set output mode to '*column*' .

-csv set output mode to '*csv*' .

-html set output mode to '*HTML*' .

-init *filename*
 read/process commands from the *filename*.

-line set output mode to '*line*' .

-list set output mode to '*list*' .

-separator '*x*'
 set output field separator (*()*).

-nullvalue '*text*'
 set text string for NULL values.

-version show SQLite version.

FILENAME
 database file name.

SQL SQL statements.

ENVIRONMENT

See HTML SQLite documentation in <https://sqlite.org/cli.html>.

EXIT STATUS

See HTML SQLite documentation in <https://sqlite.org/cli.html>.

FILES

See HTML SQLite documentation in <https://sqlite.org/docs.html>.

lib/\$OSID/includes/sqlite3.h
 include file.

lib/\$OSID/includes/sqlite3ext.h
include file.

EXAMPLES

See HTML SQLite documentation in <https://sqlite.org/cli.html>.

SEE ALSO

[edrcintro\(1\)](#), [thttpd\(3\)](#), <http://www.sqlite.org>, <https://sqlite.org/cli.html>, <https://sqlite.org/docs.html>, <https://sqlite.org/lang.html>, <https://www.sqlitetutorial.net/>

NOTES

Parts of this manual page were extracted from the **sqlite3** HTML documentation distributed with the SQLite version 3.6.3, written by "The SQLite Development Team" led by D. Richard Hipp, Dan Kennedy and Shane Harrelson.

BUGS

-

AUTHOR

sqlite was developed by D. Richard Hipp, Dan Kennedy and Shane Harrelson and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports related to the integration to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ssh-exec – execute command provided thru stdin using ssh on remote host

SYNOPSIS

edrc/bin/ssh-exec [**-h**]

ssh-exec [**-b**] **-u** *user@hostname* < *commands*

ssh-exec [**-b**] **-u** *user* **-t** *hostname* < *commands*

ssh-exec **-p** [**-e** | **-l** *localuser*] **-u** *user@hostname*

ssh-exec **-p** [**-e** | **-l** *localuser*] **-u** *user* **-t** *hostname*

ssh-exec [**-s** | **-v**][**-n**][**-b**][**-e** | **-l** *localuser*] **-u** *user@hostname* < *commands*

ssh-exec [**-s** | **-v**][**-n**][**-b**][**-e** | **-l** *localuser*] **-u** *user* **-t** *hostname* < *commands*

AVAILABILITY

WA2L/edrc

DESCRIPTION

execute *commands*, a sequence of commands or a whole script provided thru **stdin** to **ssh-exec** on a remote system using the **SSH** protocol.

Prior to the remote command invocation it is checked if the remote target system with *hostname* is up and if it is possible to connect using the provided *user* to the remote target system.

In addition to the commands it is supported to submit answers to specific outputs (when it is not possible to do it using simple input redirection) or an **except(3)** script without the need to write extra "wrapper" scripts.

[COMMANDS] Execute commands:

Simple command invocation on a target system:

```
ssh-exec -u user@target <<EOM
ps -ef | grep oracle
EOM
```

or

```
ssh-exec -u user@target <<EOM
[COMMANDS]
ps -ef | grep oracle
EOM
```

The **[COMMANDS]** section keyword is optional as long as it is the first section provided to **ssh-exec**.

Command invocation with input redirection of needed command inputs:

```
ssh-exec -u user@target <<EOM

restore-data <<EOI
/dat/backup/acme-001/2020-01-10/etc/hosts
/etc/hosts
EOI

EOM
```

[ANSWERS] Providing answers to a tty command:

Remote control an interactive command that reads the input from a **tty** and not from **stdin**:

```
ssh-exec -u user@target <<EOM

[COMMANDS]
set-license-key

[ANSWERS]
;Input new license key;;VcB542RSXtMzM\r;
;Repeat new license key;;VcB542RSXtMzM\r;

EOM
```

Behind the scenes the provided answers in the **[ANSWERS]** section are translated automatically into an ad-hoc **expect** script, as:

```
expect "Input new license key:"
send -- "VcB542RSXtMzM\r"

expect "Repeat new license key:"
send -- "VcB542RSXtMzM\r"

expect eof
```

that is used to remote control the command.

This helps to efficiently handle the most cases of remote command control where the needed **expect** definition would be a **expect ... -> send -- ...** sequence.

The **[ANSWERS]** section has the format:

```
:Input prompt:Answer:
```

where the first character is treated as separator between the *Input prompt* and the *Answer* fields.

Therefore, if the *Input prompt* or the *Answer* contains the field separator character (the colon : in this example) the specification

```
;Input prompt;Answer;
```

is identical to the specification above.

[EXPECT] Providing an expect script to a tty command:

Remote control an interactive command that reads the input from a **tty** and not from **stdin**.

The **expect** script has to be provided only if the automatically generated script using the **[ANSWERS]** section is not sufficient to the needs to control a command.

```
ssh-exec -u user@target <<EOM

[COMMANDS]
set-license-key

[EXPECT]
expect "Input new license key:"
send -- "VcB542RSXtMZM\r"

expect "Repeat new license key:"
send -- "VcB542RSXtMZM\r"

expect eof

EOM
```

OPTIONS

- h** usage message.
- p** only probe connection to target *hostname* with *user* without reading commands from **stdin**.
- e** use **EDRC_USER** as defined in **etc/ssh-exec.cfg** (default=**edrc**) as local user.

When **-e** is used or the **EDRC_USER** (default=**edrc**) is used in the **-l localuser** option, the private keys as defined in **var/connection/security/edrc/OpenSSH/default/default/id_dsa** and **var/connection/security/edrc/OpenSSH/default/default/id_rsa** are used to connect to the remote *hostname*.

Prerequisite to a working connection is that the contents of the files **var/connection/security/edrc/OpenSSH/default/default/id_dsa.pub** and **var/connection/security/edrc/OpenSSH/default/default/id_rsa.pub** are appended to the **~user/.ssh/authorized_keys** file on the target *hostname*.

To efficiently add the public keys to **authorized_keys** files on hosts where the WA2L/edrc package is installed, use the **ssh-keyadd(1m)** command.

- s** output (to **stderr**) of commands and exit without executing any commands on *hostname*.
- v** verbose output (to **stderr**) of commands to be sent to target *hostname*.

- n** do not probe connection before executing the provided *commands*.
- b** eliminate the output of a banner of a remote system (when not providing a [ANSWERS] section).
- l *localuser***
use the local user *localuser* to initiate the connection. Default local user is **root**.

When using this option the private **SSH** keys of the specified *localuser* (as defined in *~localuser/.ssh/id_dsa* and *~localuser/.ssh/id_rsa* files) on the local system are used to connect to the target *hostname*.

See **-e** option description for additional information.
- u *user*** remote user to connect on the target system.
- t *hostname***
target system *hostname* to execute the *commands*.
- u *user@hostname***
remote *user* to connect on the target system *hostname*.

commands commands to be executed on the target *hostname*.

Commands can be read from file using the input redirection:

```
ssh-exec options < commands
```

providing the file contents thru the pipe:

```
cat commands | ssh-exec options
```

or the 'in here' mechanism in scripts without the creation of an extra file:

```
ssh-exec options <<EOM
command_1
command_2
command_n
EOM
```

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 1** remote system is not up.

- 2** cannot connect to *hostname* with user *user*.
- 4** usage printed.
- 5** the execution has been aborted using *Ctrl+C*.
- 6** the configuration file **ssh-exec.cfg** does not exist.
- 7** errors in [ANSWERS] section. If you receive this error, use the **-v** to see more details.
- 8** command was started with the **-s** option.
- 11** temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

etc/ssh-exec.cfg

configuration file of **ssh-exec**, see **ssh-exec.cfg(4)** for more information.

var/log/ssh-exec.log

default logfile of **ssh-exec**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **filedist(1)**, **hostlist(3)**, **input(3)**, **input_targets(3)**, **rcmd(1)**, **remote_shell(3)**, **remote_copy(3)**, **resolve_targetlist(3)**, **ssh-exec.cfg(4)**, **ssh-keygen(1)**, **ssh-keyadd(1m)**, **syspoll(1)**

NOTES

If you need to invoke ad-hoc commands on all systems in your environment (needs WA2L/edrc installed) most efficiently use the **rcmd(1)** command.

To poll a list of systems for reporting data most efficiently, use the **syspoll(1)** command.

BUGS

-

AUTHOR

ssh-exec was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ssh-exec.cfg – configuration file for ssh-exec

SYNOPSIS

edrc/etc/ssh-exec.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **ssh-exec** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **ssh-exec**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

EDRC_USER

The home of this user is considered as the root of the WA2L/edrc installation. See **edrc-setup**(1m) for information about user settings needed by WA2L/edrc.

Example: EDRC_USER=edrc

Default: EDRC_USER=edrc

START_USER

User that is allowed to start **ssh-exec**.

This setting is of relevance if the user-id of the starting user is different from the local user used (**-l localuser** option) to connect to the remote system.

Example: **START_USER=eroot**

Default: **START_USER=root**

CONNECTION_TIMEOUT

ssh connection timeout in seconds.

Specifies the timeout (in seconds) used when connecting to the SSH server, instead of using the default system TCP timeout.

This value is used only when the target is down or really unreachable, not when it refuses the connection.

Example: **CONNECTION_TIMEOUT=2**

Default: **CONNECTION_TIMEOUT=4**

PROBE_COMMAND

Command to invoke on the remote system to probe a ssh connection with the given remote user.

The chosen command must be available on all target systems.

If it is needed to set the **PROBE_COMMAND** dynamically the environment variables **\$REMOTEUSER** and/or **\$TARGET** can be used to do so.

The **\$REMOTEUSER** variable contains the user name as given on the command line in lower case; the **\$TARGET** variable contains the target system as given on the command line also in lower case.

Example:

```
case $TARGET in
  acme007) PROBE_COMMAND="uname -n" ;;
  alpha60) PROBE_COMMAND="sysname" ;;
  joshua)  PROBE_COMMAND="identity" ;;
  hal9000) PROBE_COMMAND="avatar"  ;;
  bomb20) PROBE_COMMAND="WHO"      ;;
  *)      PROBE_COMMAND="hostname" ;;
esac
```

Example: **PROBE_COMMAND=hostname**

Default: **PROBE_COMMAND=hostname**

SEE ALSO

edrcintro(1), **edrcsetup(1m)**, **ssh-exec(1)**

NOTES

-

BUGS

-

AUTHOR

ssh-exec.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ssh-keyadd – add ssh keys needed for WA2L/edrc to authorized_keys file

SYNOPSIS

edrc/bin/ssh-keyadd [**-h**]

ssh-keyadd [**-y**] [**-u** *user*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Add SSH public keys to a users (normally root's) **authorized_keys** file(s).

This command is used to help to efficiently add the needed SSH keys to ensure the connectivity for WA2L/edrc needed for the commands as **remote_shell**, **remote_copy**, **filedist**, **rcmd** and the **edrc** trunk functionality.

Furthermore the SSH daemon configuration file is modified that way that it is allowed to connect as root. To make the configuration change active, the HUP signal is sent to the SSH daemon.

With **ssh-keyadd** it is also possible to get an overview of the current SSH key situation due to the fact, that the tool gives an easy to read consolidated view of the applied keys.

OPTIONS

-h usage message.

-y answer to "yes" to all interactive questions.

-u *user* username if it is different from the default (root).

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 configuration file does not exist.

- 4 Usage printed.
- 6 **ssh-keyadd** aborted pressing <Ctrl>+<C>.

FILES

edrc/etc/ssh-keyadd.cfg
ssh-keyadd configuration file.

edrc/etc/ssh-keyadd.pub
 global public keys to be added.

/etc/ssh/sshd_config
 configuration file of the SSH daemon.

/etc/ssh/etc/sshd_config
 configuration file of the SSH daemon.

edrc/var/connection/security/[<EDRC_USER>|<user>]/OpenSSH/*/[default|<hostname>]/*.pub
 local public keys to be added.

~<user>/.ssh/authorized_keys
 SSH authorized keys file of the user. See also **ssh-keyadd.cfg** for more information.

~<user>/.ssh/authorized_keys2
 SSH authorized keys file of the user. See also **ssh-keyadd.cfg** for more information.

EXAMPLES

-

SEE ALSO

edrcintro(1), **remote_copy**(3), **remote_shell**(3), **ssh-exec**(1), **ssh-keyadd.cfg**(4), **ssh-keyadd.pub**(4), **ssh**(1), **sshd**(1m), **sshd_config**(4), **ssh-keygen**(1)

NOTES

-

BUGS

-

AUTHOR

ssh-keyadd was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ssh-keyadd.cfg – configuration file for ssh-keyadd

SYNOPSIS

edrc/etc/ssh-keyadd.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **ssh-keyadd** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**SSHD_CONFIGFILE**

List of SSH configuration files.

Example: SSHD_CONFIGFILE="/etc/ssh/sshd_config /opt/ssh/etc/sshd_config"

Default: SSHD_CONFIGFILE="/etc/ssh/sshd_config /opt/ssh/etc/sshd_config"

SSHD_PROCESSNAME

List of SSH process names.

Example: SSHD_PROCESSNAME="/usr/sbin/sshd /opt/ssh/sbin/sshd /usr/lib/ssh/sshd"

Default: SSHD_PROCESSNAME="/usr/sbin/sshd /opt/ssh/sbin/sshd /usr/lib/ssh/sshd"

SSHD_PIDFILE

List of SSH PID file names.

Example: SSHD_PIDFILE="/var/run/sshd.pid /var/run/sshd.init.pid"

Default: SSHD_PIDFILE="/var/run/sshd.pid /var/run/sshd.init.pid"

AUTHORIZED_KEYS_FILE

List of authorized key files in \$HOME/.ssh directory.

Example: AUTHORIZED_KEYS_FILE="authorized_keys"

Default: AUTHORIZED_KEYS_FILE="authorized_keys authorized_keys2"

LOG

Log output dir of ssh-keyadd. If you specify a relative path name the path is relative to the root of the EDRC installation.

Example: LOG=var/log

Default: LOG=var/log

EDRC_USER

Name of edrc user.

Example: EDRC_USER=edrc

Default: EDRC_USER=edrc

SEE ALSO

edrcintro(1), **ssh-keyadd(1m)**, **ssh-keyadd.pub(4)**

NOTES

-

BUGS

-

AUTHOR

ssh-keyadd.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ssh-keyadd.pub – global SSH public keys

SYNOPSIS

edrc/etc/ssh-keyadd.pub

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **ssh-keyadd** command and contains all global public SSH keys which are used to connect from a central management environment to all environments.

FILEFORMAT

The fileformat is a plain file containing public keys.

Rows starting with a **#** are considered as comments.

OPTIONS

-

EXAMPLES

Example configuration file containing two global SSH keys to allow the administration environment 'Highlander' to access the systems in the environment.

One key is on one line without returns and goes from 'ssh-...' until 'edrc@Highlander'.

```
#
# ssh-keyadd.pub - global SSH public keys
#
# [00] 19.09.2010 CWa Initial Version
#
ssh-dss AATAB3NzaC1kc3MAZACBANTZUzTJIZFxNiZprstBIsgXgF
wk/OrIvRMUlkJIrJRBqITLdxF8sQllWMjhrufv97yd9oUAnNv0j9Yj
BYNdT/ZvHn9VadZYZAioHQ5N+dvX8EbZWbLtEYLi49Gwfkf54Tzs0
ZQLUWrRC3yF+EI0bpAhzd6gBiqHK+AcoKTZ+58h8cQAAIEAoSqZXw
jZJfXPJ2Q0eqMbqeTAXT3oML5byrxw+fnANUVIkc8nlUDmQQUAg3Eg
mO7cSya1FPXdkpqefv4vtninFndj8muwsRhZGXgYFwRggvtaV/rtn3
Fptdk7NZD6/XCpM9JXJCMubIhhFbQ= edrc@Highlander
ssh-rsa ATAAB3NzaC1yc2EAAAABIwAAAQEAvkdixAj3Jd9WfuiTSu
```

```
771Zxi3MnN+9VBxuCXKuEneX0e5E8BTc0Nx4XXF12qeOTTfBVZVJUv
Q55ohDhLQ/7SKLF7B1na4JYAONnqLpR/A+cFTI9jQrPHudEhPwS80x
x65jSCBOMq+ydZtFLLFnd+HiX/5rzK86QXC7HCcnfXezr+78jcpBgW
DpQ== edrc@Highlander
```

SEE ALSO

edrcintro(1), **ssh-keyadd**(1m), **ssh-keyadd.cfg**(4)

NOTES

-

BUGS

-

AUTHOR

ssh-keyadd.pub was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sshpass – noninteractive ssh password provider

SYNOPSIS

sshpass [-f*filename*|-d*num*|-p*password*|-e] [*options*] *command arguments*

DESCRIPTION

This manual page documents the **sshpass** command.

sshpass is a utility designed for running **ssh** using the mode referred to as "keyboard-interactive" password authentication, but in non-interactive mode.

ssh uses direct TTY access to make sure that the password is indeed issued by an interactive keyboard user. Sshpass runs ssh in a dedicated tty, fooling it into thinking it is getting the password from an interactive user.

The command to run is specified after sshpass' own options. Typically it will be "ssh" with arguments, but it can just as well be any other command. The password prompt used by ssh is, however, currently hard-coded into sshpass.

Options

If no option is given, sshpass reads the password from the standard input. The user may give at most one alternative source for the password:

-p*password*

The password is given on the command line. Please note the section titled "**SECURITY CONSIDERATIONS**".

-f*filename*

The password is the first line of the file *filename*.

-d*number*

number is a file descriptor inherited by sshpass from the runner. The password is read from the open file descriptor.

-e

The password is taken from the environment variable "SSHPASS".

-P

Set the password prompt. Sshpass searched for this prompt in the program's output to the TTY as an indication when to send the password. By default sshpass looks for the string "assword:" (which matches both "Password:" and "password:"). If your client's prompt does not fall under either of these, you can override the default with this option.

-v

Be verbose. sshpass will output to stderr information that should help debug cases where the connection hangs, seemingly for no good reason.

SECURITY CONSIDERATIONS

First and foremost, users of sshpass should realize that ssh's insistence on only getting the password interactively is not without reason. It is close to impossible to securely store the password, and users of sshpass should consider whether ssh's public key authentication provides the same end-user experience, while involving less hassle and being more secure.

The -p option should be considered the least secure of all of sshpass's options. All system users can see the password in the command line with a simple "ps" command. Sshpass makes a minimal attempt to hide the password, but such attempts are doomed to create race conditions without actually solving the problem. Users of sshpass are encouraged to use one of the other password passing techniques, which are all more secure.

In particular, people writing programs that are meant to communicate the password programatically are encouraged to use an anonymous pipe and pass the pipe's reading end to sshpass using the -d option.

RETURN VALUES

As with any other program, sshpass returns 0 on success. In case of failure, the following return codes are used:

- 1 Invalid command line argument
- 2 Conflicting arguments given
- 3 General runtime error
- 4 Unrecognized response from ssh (parse error)
- 5 Invalid/incorrect password
- 6 Host public key is unknown. sshpass exits without confirming the new key.

In addition, ssh might be complaining about a man in the middle attack. This complaint does not go to the tty. In other words, even with sshpass, the error message from ssh is printed to standard error. In such a case ssh's return code is reported back. This is typically an unimaginative (and non-informative) "255" for all error cases.

EXAMPLES

Run rsync over SSH using password authentication, passing the password on the command line:

```
rsync --rsh='sshpass -p 12345 ssh -l test' host.example.com:path .
```

To do the same from a bourne shell script in a marginally less exposed way:

```
SSHPASS=12345 rsync --rsh='sshpass -e ssh -l test' host.example.com:path .
```

BUGS

Sshpass is in its infancy at the moment. As such, bugs are highly possible. In particular, if the password is read from stdin (no password option at all), it is possible that some of the input aimed to be passed to ssh will be read by sshpass and lost.

Sshpass utilizes the **pty(7)** interface to control the TTY for ssh. This interface, at least on Linux, has a misfeature where if no slave file descriptors are open, the master pty returns **EIO**. This is the normal behavior, except a slave pty may be born at any point by a program opening **/dev/tty**. This makes it impossible to reliably wait for events without consuming 100% of the CPU.

Over the various versions different approaches were attempted at solving this problem. Any given version of sshpass is released with the belief that it is working, but experience has shown that these things do, occasionally, break. This happened with OpenSSH version 5.6. As of this writing, it is believed that sshpass is, again, working properly.

NAME

stat – display stat() information of a file

SYNOPSIS

edrc/bin/stat [**-h**]

stat -t

stat [**-L**] **-s** *stat_field* **-f** *file*

cat *filelist* | **stat** [**-L**] **-s** *stat_field* **-f** -

stat [**-L**] **-s** *stat_field* **-f** - < *filelist*

AVAILABILITY

WA2L/edrc

DESCRIPTION

This command returns the file status information returned by the **stat(2)** system call.

OPTIONS

-h usage message.

-t print the CSV header of all columns as returned when invoking **-s all**.

-L follow symlinks.

-s *stat_field*

as one of the following:

dev device number of the filesystem where the file resides.

ino inode number of the file.

mode file mode (type and permission) as numeric representation.

perm file permission in octal representation.

nlink number of hard links to the file.

| | |
|----------------|--|
| uid | numeric user ID of the file owner. |
| gid | numeric group ID of the file owner. |
| rdev | device identifier. This only applies for special files. |
| size | total size of the file in bytes. |
| atime | last access time since the epoch. |
| atimel | last access time in local time representation printed in military format YYYY-MM-DD hh:mm:ss (e.g. 2010-08-28 11:55:59). |
| mtime | last modify time since the epoch. |
| mtimel | last modify time in local time representation printed in military format YYYY-MM-DD hh:mm:ss (e.g. 2010-08-28 11:55:59). |
| ctime | inode change time (*not* the creation time!) since the epoch. |
| ctimel | inode change time (*not* the creation time!) in local time representation printed in military format YYYY-MM-DD hh:mm:ss (e.g. 2010-08-28 11:55:59). |
| blksize | preferred block size for file system I/O. |
| blocks | actual number of blocks allocated. |
| type | type of the file: |
| f | File is a plain file. |
| d | File is a directory. |
| l | File is a symbolic link. |
| p | File is a pipe (FIFO). |
| S | File is a socket. |
| b | File is a block special file. |
| c | File is a character special file. |
| t | File is opened by a tty. |

T File is a text file.

B File is a binary file.

all print all **stat** fields including the *file* as a CSV output. The order of the output fields is:
DEV;INO;MODE;PERM;NLINK;UID;GID;RDEV;SIZE;ATIME;MTIME;CTIME;BLK-SIZE;BLOCKS;FILE;ATIMEL;MTIMEL;CTIMEL;TYPE;.

-f file filename of the file for which the stat information should be printed. If **-** is given as **file stdin** is read for a list of files.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 1** file specified is not readable.
- 2** operating system is not supported, yet. See **osid(3)** if you get this error.
- 4** usage printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), osid(3), print_list(3), select_columns(3), sparse(3), stat(2), timer(1)

NOTES

-

BUGS

when using **-L** the type output is **l** and not the type of the file the symlink is pointing to.

AUTHOR

stat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

streamcat – continuous (unbuffered) output of a stream

SYNOPSIS

edrc/lib/streamcat < FIFO

streamcat < file

command | **streamcat**

AVAILABILITY

WA2L/edrc

DESCRIPTION

continuous unbuffered output of an input stream received thru **stdin** to **stdout**.

OPTIONS

file stream to be displayed.

command stream to be displayed.

FIFO stream to be displayed.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **cat(1)**, **tee(1)**, **xmore(1)**, **xtee(1)**

NOTES

-

BUGS

-

AUTHOR

streamcat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

strupr, strlwr – additional string functions

SYNOPSIS

```
#include "strings.h"
```

```
char *strupr(char *string);
```

```
char *strlwr(char *string);
```

AVAILABILITY

WA2L/edrc

DESCRIPTION

strings.h provides additional string functions not provided in **<string.h>**.

strupr()

return provided string in upper case.

Example:

```
printf("upper case string: '%s'\n", strupr("a String to be converted"));
```

Output:

```
upper case string: 'A STRING TO BE CONVERTED'
```

strlwr()

return provided string in lower case.

Example:

```
printf("lower case string: '%s'\n", strlwr("a String to be converted"));
```

Output:

```
lower case string: 'a string to be converted'
```

RETURN VALUE

the return value of **strupr()** and **strlwr()** is a pointer to the converted variant of the given string.

ENVIRONMENT

-

FILES

`lib/$OSID/includes/strings.h`

EXAMPLES

-

SEE ALSO

`edrcintro(1)`, <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>, `checkopt.h(3)`, `osid(3)`, `program.h(3)`, `utility.h(3)`, `wa2lc(3)`

NOTES

-

BUGS

-

AUTHOR

`strings.h` was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

strace – trace system calls and signals

SYNOPSIS

```
strace [ -dffhiqrtttTvxx ] [ -acolumn ] [ -eexpr ] ... [ -ofile ] [ -ppid ] ... [ -sstrsize ] [ -uusername ] [ command [ arg ... ] ]
```

```
strace -c [ -eexpr ] ... [ -Ooverhead ] [ -Ssortby ] [ command [ arg ... ] ]
```

DESCRIPTION

In the simplest case **strace** runs the specified *command* until it exits. It intercepts and records the system calls which are called by a process and the signals which are received by a process. The name of each system call, its arguments and its return value are printed on standard error or to the file specified with the **-o** option.

strace is a useful diagnostic, instructional, and debugging tool. System administrators, diagnosticians and trouble-shooters will find it invaluable for solving problems with programs for which the source is not readily available since they do not need to be recompiled in order to trace them. Students, hackers and the overly-curious will find that a great deal can be learned about a system and its system calls by tracing even ordinary programs. And programmers will find that since system calls and signals are events that happen at the user/kernel interface, a close examination of this boundary is very useful for bug isolation, sanity checking and attempting to capture race conditions.

Each line in the trace contains the system call name, followed by its arguments in parentheses and its return value. An example from stracing the command “cat /dev/null” is:

```
open("/dev/null", O_RDONLY) = 3
```

Errors (typically a return value of -1) have the errno symbol and error string appended.

```
open("/foo/bar", O_RDONLY) = -1 ENOENT (No such file or directory)
```

Signals are printed as a signal symbol and a signal string. An excerpt from stracing and interrupting the command “sleep 666” is:

```
sigsuspend([] <unfinished ...>
--- SIGINT (Interrupt) ---
+++ killed by SIGINT +++
```

Arguments are printed in symbolic form with a passion. This example shows the shell performing “>>xyzy” output redirection:

```
open("xyzy", O_WRONLY|O_APPEND|O_CREAT, 0666) = 3
```

Here the three argument form of open is decoded by breaking down the flag argument into its three bitwise-OR constituents and printing the mode value in octal by tradition. Where traditional or native usage differs from ANSI or POSIX, the latter forms are preferred. In some cases, **strace** output has proven to be more readable than the source.

Structure pointers are dereferenced and the members are displayed as appropriate. In all cases arguments are formatted in the most C-like fashion possible. For example, the essence of the command “ls -l /dev/null” is captured as:

```
lstat("/dev/null", {st_mode=S_IFCHR|0666, st_rdev=makedev(1, 3), ...}) = 0
```

Notice how the ‘struct stat’ argument is dereferenced and how each member is displayed symbolically. In particular, observe how the st_mode member is carefully decoded into a bitwise-OR of symbolic and

numeric values. Also notice in this example that the first argument to `lstat` is an input to the system call and the second argument is an output. Since output arguments are not modified if the system call fails, arguments may not always be dereferenced. For example, retrying the “`ls -l`” example with a non-existent file produces the following line:

```
lstat("/foo/bar", 0xb004) = -1 ENOENT (No such file or directory)
```

In this case the porch light is on but nobody is home.

Character pointers are dereferenced and printed as C strings. Non-printing characters in strings are normally represented by ordinary C escape codes. Only the first *strsize* (32 by default) bytes of strings are printed; longer strings have an ellipsis appended following the closing quote. Here is a line from “`ls -l`” where the **getpwuid** library routine is reading the password file:

```
read(3, "root::0:0:System Administrator:/...", 1024) = 422
```

While structures are annotated using curly braces, simple pointers and arrays are printed using square brackets with commas separating elements. Here is an example from the command “`id`” on a system with supplementary group ids:

```
getgroups(32, [100, 0]) = 2
```

On the other hand, bit-sets are also shown using square brackets but set elements are separated only by a space. Here is the shell preparing to execute an external command:

```
sigprocmask(SIG_BLOCK, [CHLD TTOU], []) = 0
```

Here the second argument is a bit-set of two signals, `SIGCHLD` and `SIGTTOU`. In some cases the bit-set is so full that printing out the unset elements is more valuable. In that case, the bit-set is prefixed by a tilde like this:

```
sigprocmask(SIG_UNBLOCK, ~[], NULL) = 0
```

Here the second argument represents the full set of all signals.

OPTIONS

- c** Count time, calls, and errors for each system call and report a summary on program exit.
- d** Show some debugging output of **strace** itself on the standard error.
- f** Trace child processes as they are created by currently traced processes as a result of the **fork(2)** system call. The new process is attached to as soon as its pid is known (through the return value of **fork(2)** in the parent process). This means that such children may run uncontrolled for a while (especially in the case of a **vfork(2)**), until the parent is scheduled again to complete its **(v)fork(2)** call. If the parent process decides to **wait(2)** for a child that is currently being traced, it is suspended until an appropriate child process either terminates or incurs a signal that would cause it to terminate (as determined from the child’s current signal disposition).
- ff** If the **-o filename** option is in effect, each processes trace is written to *filename.pid* where pid is the numeric process id of each process.
- F** Attempt to follow **vforks**. (On SunOS 4.x, this is accomplished with some dynamic linking trickery. On Linux, it requires some kernel functionality not yet in the standard kernel.) Otherwise, **vforks** will not be followed even if **-f** has been given.
- h** Print the help summary.

- i** Print the instruction pointer at the time of the system call.
- q** Suppress messages about attaching, detaching etc. This happens automatically when output is redirected to a file and the command is run directly instead of attaching.
- r** Print a relative timestamp upon entry to each system call. This records the time difference between the beginning of successive system calls.
- t** Prefix each line of the trace with the time of day.
- tt** If given twice, the time printed will include the microseconds.
- ttt** If given thrice, the time printed will include the microseconds and the leading portion will be printed as the number of seconds since the epoch.
- T** Show the time spent in system calls. This records the time difference between the beginning and the end of each system call.
- v** Print unabbreviated versions of environment, stat, termios, etc. calls. These structures are very common in calls and so the default behavior displays a reasonable subset of structure members. Use this option to get all of the gory details.
- V** Print the version number of **strace**.
- x** Print all non-ASCII strings in hexadecimal string format.
- xx** Print all strings in hexadecimal string format.
- a column** Align return values in a specific column (default column 40).
- e expr** A qualifying expression which modifies which events to trace or how to trace them. The format of the expression is:

`[qualifier=][!]value1[,value2]`...

where *qualifier* is one of **trace**, **abbrev**, **verbose**, **raw**, **signal**, **read**, or **write** and *value* is a qualifier-dependent symbol or number. The default qualifier is **trace**. Using an exclamation mark negates the set of values. For example, **-eopen** means literally **-e trace=open** which in turn means trace only the **open** system call. By contrast, **-etrace=!open** means to trace every system call except **open**. In addition, the special values **all** and **none** have the obvious meanings.

Note that some shells use the exclamation point for history expansion even inside quoted arguments. If so, you must escape the exclamation point with a backslash.

- e trace=set** Trace only the specified set of system calls. The **-c** option is useful for determining which system calls might be useful to trace. For example, **trace=open,close,read,write** means to only trace those four system calls. Be careful when making inferences about the user/kernel boundary if only a subset of system calls are being monitored. The default is **trace=all**.
- e trace=file** Trace all system calls which take a file name as an argument. You can think of this as an abbreviation for **-e trace=open,stat,chmod,unlink,...** which is useful to seeing what files the process is referencing. Furthermore, using the abbreviation will ensure that you don't accidentally forget to include a call like **lstat** in the list. Betchya woulda forgot that one.
- e trace=process** Trace all system calls which involve process management. This is useful for watching the fork, wait, and exec steps of a process.
- e trace=network** Trace all the network related system calls.
- e trace=signal** Trace all signal related system calls.
- e trace=ipc** Trace all IPC related system calls.

- e abbrev=*set***
Abbreviate the output from printing each member of large structures. The default is **abbrev=all**. The **-v** option has the effect of **abbrev=none**.
- e verbose=*set***
Dereference structures for the specified set of system calls. The default is **verbose=all**.
- e raw=*set***
Print raw, undecoded arguments for the specified set of system calls. This option has the effect of causing all arguments to be printed in hexadecimal. This is mostly useful if you don't trust the decoding or you need to know the actual numeric value of an argument.
- e signal=*set***
Trace only the specified subset of signals. The default is **signal=all**. For example, **signal=!SIGIO** (or **signal=!io**) causes SIGIO signals not to be traced.
- e read=*set***
Perform a full hexadecimal and ASCII dump of all the data read from file descriptors listed in the specified set. For example, to see all input activity on file descriptors 3 and 5 use **-e read=3,5**. Note that this is independent from the normal tracing of the **read(2)** system call which is controlled by the option **-e trace=read**.
- e write=*set***
Perform a full hexadecimal and ASCII dump of all the data written to file descriptors listed in the specified set. For example, to see all output activity on file descriptors 3 and 5 use **-e write=3,5**. Note that this is independent from the normal tracing of the **write(2)** system call which is controlled by the option **-e trace=write**.
- o *filename***
Write the trace output to the file *filename* rather than to stderr. Use *filename.pid* if **-ff** is used. If the argument begins with '|' or with '!' then the rest of the argument is treated as a command and all output is piped to it. This is convenient for piping the debugging output to a program without affecting the redirections of executed programs.
- O *overhead***
Set the overhead for tracing system calls to *overhead* microseconds. This is useful for overriding the default heuristic for guessing how much time is spent in mere measuring when timing system calls using the **-c** option. The accuracy of the heuristic can be gauged by timing a given program run without tracing (using **time(1)**) and comparing the accumulated system call time to the total produced using **-c**.
- p *pid***
Attach to the process with the process ID *pid* and begin tracing. The trace may be terminated at any time by a keyboard interrupt signal (CTRL-C). **strace** will respond by detaching itself from the traced process(es) leaving it (them) to continue running. Multiple **-p** options can be used to attach to up to 32 processes in addition to *command* (which is optional if at least one **-p** option is given).
- s *strsize***
Specify the maximum string size to print (the default is 32). Note that filenames are not considered strings and are always printed in full.
- S *sortby***
Sort the output of the histogram printed by the **-c** option by the specified criterion. Legal values are **time**, **calls**, **name**, and **nothing** (default **time**).
- u *username***
Run command with the user ID, group ID, and supplementary groups of *username*. This option is only useful when running as root and enables the correct execution of **setuid** and/or **setgid** binaries. Unless this option is used **setuid** and **setgid** programs are executed without effective privileges.

SETUID INSTALLATION

If **strace** is installed **setuid** to root then the invoking user will be able to attach to and trace processes owned by any user. In addition **setuid** and **setgid** programs will be executed and traced with the correct effective privileges. Since only users trusted with full root privileges should be allowed to do these things, it only makes sense to install **strace** as **setuid** to root when the users who can execute it are restricted to those users who have this trust. For example, it makes sense to install a special version of **strace** with mode 'rwsr-xr--', user **root** and group **trace**, where members of the **trace** group are trusted users. If you do use this feature, please remember to install a non-**setuid** version of **strace** for ordinary users to use.

SEE ALSO

ptrace(2), **proc(4)**, **time(1)**, **trace(1)**, **truss(1)**

NOTES

It is a pity that so much tracing clutter is produced by systems employing shared libraries.

It is instructive to think about system call inputs and outputs as data-flow across the user/kernel boundary. Because user-space and kernel-space are separate and address-protected, it is sometimes possible to make deductive inferences about process behavior using inputs and outputs as propositions.

In some cases, a system call will differ from the documented behavior or have a different name. For example, on System V-derived systems the true **time(2)** system call does not take an argument and the **stat** function is called **xstat** and takes an extra leading argument. These discrepancies are normal but idiosyncratic characteristics of the system call interface and are accounted for by C library wrapper functions.

On some platforms a process that has a system call trace applied to it with the **-p** option will receive a **SIGSTOP**. This signal may interrupt a system call that is not restartable. This may have an unpredictable effect on the process if the process takes no action to restart the system call.

BUGS

Programs that use the *setuid* bit do not have effective user ID privileges while being traced.

A traced process ignores **SIGSTOP** except on SVR4 platforms.

A traced process which tries to block **SIGTRAP** will be sent a **SIGSTOP** in an attempt to force continuation of tracing.

A traced process runs slowly.

Traced processes which are descended from *command* may be left running after an interrupt signal (**CTRL-C**).

On Linux, exciting as it would be, tracing the *init* process is forbidden.

The **-i** option is weakly supported.

HISTORY

strace The original **strace** was written by Paul Kranenburg for SunOS and was inspired by its trace utility. The SunOS version of **strace** was ported to Linux and enhanced by Branko Lankester, who also wrote the Linux kernel support. Even though Paul released **strace** 2.5 in 1992, Branko's work was based on Paul's **strace** 1.5 release from 1991. In 1993, Rick Sladkey merged **strace** 2.5 for SunOS and the second release of **strace** for Linux, added many of the features of **truss(1)** from SVR4, and produced an **strace** that worked on both platforms. In 1994 Rick ported **strace** to SVR4 and Solaris and wrote the automatic configuration support. In 1995 he ported **strace** to Irix and tired of writing about himself in the third person.

PROBLEMS

Problems with **strace** should be reported to the current **strace** maintainer, Wichert Akkerman, at <wakkerma@debian.org>.

NAME

sysvi – edit a file in sysconfig repository

SYNOPSIS

edrc/bin/sysvi [**-h**]

sysvi *file* ...

svi [**-h**]

svi *file* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

edit the file in a sysconfig repository which is related to the local (active) file on the system.

If a file is in multiple repositories it can be interactively chosen in which repository the file should be edited.

OPTIONS

-h usage message.

file ... list of files to be edited in the repository.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 2 | operating system is not supported, yet. See osid (3) if you get this error. |
| 4 | usage printed. |
| 5 | command aborted pressing <Ctrl>+<C> or using kill (1). |

- 11** temporary directory could not be claimed in **/tmp**. Check the system if you get this error, it is an indicator for system intrusion.

FILES

<root_dir>/EDRC.sysconfig<.name>

settings file(s) that contains information about current active configuration(s). This file is read when using **sysconfig -p**, **syscp**, **sysvi** and **sysdiff**.

etc/exrc settings for the **vi** / **vim** editor.

EXAMPLES

-

SEE ALSO

edrcintro(1), **filelink(1)**, **sysconfig(1)**, **syscp(1)**, **sysdiff(1)**

NOTES

-

BUGS

-

AUTHOR

sysvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sw_inventory – software inventory file

SYNOPSIS

<software installation path>/..sw_inventory

AVAILABILITY

WA2L/edrc

DESCRIPTION

"software inventory file" containing all software information of software installed in the directory where created.

This file is created when using the **swvi**(1) command.

FILEFORMAT

The fileformat is a list of definitions that have the format (all on one line):

```
SOFTWARE;VERSION;INST_USER;INST_GROUP;RUN_USER;\  
RUN_GROUP;CFG_DIR;VAR_DIR;LICENSE;LIC_MGMT;WEB;\  
DOC;INST_SRC;PRODUCT;INSTANCE;COMMENT;
```

Rows starting with a # are considered as comments. Empty lines are allowed, too.

OPTIONS**SOFTWARE**

Software name

VERSION

Software version

INST_USER

Installation user

INST_GROUP

Installation group

RUN_USER

User under which the software is started

RUN_GROUP

Group under which the software is started

CFG_DIR Configuration directories spaces separated

VAR_DIR Var directories separated with spaces

LICENSE License (e.g.: GNU, 5 LTU, ...)

LIC_MGMT

License management (file, server, ...)

WEB Source of information on Internet

DOC Where to find documentation

INST_SRC

Installation Source (SW Path, CD-Name, ...)

PRODUCT

Product (Application) that uses the software

INSTANCE

Application instance (e.g.: PRODUCTION, TEST, Online, Reporting,)

COMMENT

Additional comments

EXAMPLES

-

SEE ALSO

edrcintro(1), **apply2sw_inventory(1)**, **fields2swvi(3)**, **swvi(3)**, **sw_report(3)**, **vi(1)**

NOTES

-

BUGS

-

AUTHOR

..sw_inventory was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sw_report – produce contrib.doc OUTPUT-EXAMPLE output from logfile

SYNOPSIS

edrc/bin/sw_report [**-h**]

sw_report [**-c**] [**-t** (**table** | **csv_header** | **csv**)

sw_report last

AVAILABILITY

WA2L/edrc

DESCRIPTION

create a software inventory report of a system. This command takes the **..sw_inventory** files as a base for information.

The **sw_report** travels thru the directories and searches for **..sw_inventory** files to produce the software inventory report.

The **sw_report** command must be started as **root** user.

OPTIONS

-h usage message.

-c use the cache file of previous **sw_report** runs to collect the **..sw_inventory** file information.

-t table produce a plain **ASCII** report for direct usage.

-t csv_header
 print **CSV** header row.

-t csv produce a **CSV** report (data only).

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|--|
| 0 | no error. |
| 1 | command was not started as root user. |
| 4 | usage printed. |
| 5 | command aborted. |

FILES

| | |
|---------------------------------|--|
| ..sw_inventory | software inventory file. |
| /var/tmp/sw_report.cache | cache file of sw_report containing a list of resolved ..sw_inventory files on the system. |

EXAMPLES

-

SEE ALSO

edrcintro(1), **apply2sw_inventory(1)**, **fields2swvi(3)**, **swvi(1)**, **sw_inventory(4)**,

NOTES

-

BUGS

-

AUTHOR

sw_report was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

swvi – edit/create the software inventory file

SYNOPSIS

edrc/bin/swvi [**-h**]

swvi [**-a**][**-r**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **swvi** and **sw_report** it is very easy to create a software inventory of a system without manually maintaining extra lists that document this.

With the **swvi** command the "software inventory file" is created or edited. This file (**..sw_inventory**) contains the information about the software installed in a certain directory.

A "software inventory file" can be created by everyone having the permissions to write into the directory where the software is installed to. So the task to maintain the "software inventory file" and finally the software inventory of a system is not only dependent of the system administrator which has the root password.

This "software inventory files" spread over many directories on the system are collected with the **sw_report** command to generate either a plain ASCII- or a CSV report of all installed software.

To place this information into the same directory as the software is installed has the advantage that it is moved/copied if the directory containing the software is moved/copied to an other location. Furthermore it is more likely updated then a manually maintained list residing on some documentation server.

To create a new "software inventory file" you have to change to the directory where the software is installed and call the **swvi** command. An interactive dialog will query you information needed. All information except the **SOFTWARE**, **VERSION**, **INST_USER** and **INST_GROUP** fields are optional.

If there is already an existing **..sw_inventory** file in the directory, the current file is loaded into **vi** for editing.

If a second software is installed into the same directory (what is often the case for **/usr/local**) you can invoke **swvi -a** to append a new record to the file answering the interactive dialog.

Be aware that using the **-a** option will ensure that only one identical **SOFTWARE**, **VERSION** pair exists in one **..sw_inventory** file.

The "software inventory file" is created with the permissions '644' and the user/group of the creating user. If the 'root' user invokes **swvi** the user/group setting of the overlying directory are set to ensure that the software owner (= the user ID that has been used to install the software) is able to modify the **..sw_inventory** file in the future.

The following fields will be queried (see also output of **swvi -h**):

* = required entries

SOFTWARE*

Software name

VERSION*

Software version

INST_USER*

Installation user

INST_GROUP*

Installation group

RUN_USER

User under which the software is started

RUN_GROUP

Group under which the software is started

CFG_DIR Configuration directories spaces separated

VAR_DIR Var directories separated with spaces

LICENSE License (e.g.: GNU, 5 LTU, ...)

LIC_MGMT

License management (file, server, ...)

WEB Source of information on Internet

DOC Where to find documentation

INST_SRC

Installation Source (SW Path, CD-Name, ...)

PRODUCT

Product (Application) that uses the software

INSTANCE

Application instance (e.g.: PRODUCTION, TEST, Online, Reporting,)

COMMENT

Additional comments

OPTIONS

- "** create a new **..sw_inventory** file if it does not exist in the current directory or load an existing **..sw_inventory** file into **vi** for editing.
- h** usage message.
- a** add a new record to an existing **..sw_inventory** file using the interactive dialog.
- r** remove (clear) the contents of an already existing **..sw_inventory** file. This option is rarely used in manual invocation, but is useful if you perform automated installations. For automated installations see also **apply2sw_inventory(1)**.

EXIT STATUS

- 0** no error.
- 1** no write permission to current working directory.
- 2** operating system not supported. See **osid(3)** if you get this error.
- 4** usage displayed.
- 5** command aborted with Ctrl+C.

FILES

- ..sw_inventory** "software inventory file" containing all software information of software installed in the directory where created.
- etc/exrc** settings for the **vi** / **vim** editor.

SEE ALSO

edrcintro(1), **sw_report(1)**, **sw_inventory(4)**, **apply2sw_inventory(1)**, **fields2swvi(3)**, **vi(1)**, **duvi(1)**

NOTES

To see how to document (label) data and other directories efficiently, see **duvi(1)**.

You might hear the argument, that maintaining a software inventory that way is not needed at all, due to the fact that the operating system provides this information via the built in software package management mechanism. But from experience, this is not sufficient for the most systems. Many software is distributed in a variety of formats, and I personally know not a single system administrator which creates for *each* piece of software that is installed a software package that is recognized by the operating system (rpm for Linux, SW-Depots for HP-UX, pkg for Solaris, ...).

But at latest, when the date of migrating the system (and all installed software and applications) to a newer one comes, the big worries/questions/issues/no-go's/show-stoppers, however you like to name it, about not knowing what is installed on the current system and a not up-to-date software inventory, will come up. And this is one moment where the usage **swvi** and **sw_report** really pays back.

BUGS

-

AUTHOR

swvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

symlink – create symlink *link* → *original*

SYNOPSIS

edrc/bin/symlink [**-h**]

symlink [**-l**] *original link*

AVAILABILITY

WA2L/edrc

DESCRIPTION

the **symlink** command is used to create a symbolic link to an original destination, basically as the '**ln -s original link**' command does.

If the *link* item already exists, it is removed prior to the symbolic link creation. To avoid data deletion, the **-l** switch can be used to ensure, that the *link* already exists and is also a symbolic link.

The main purpose of **symlink** is to toggle a symbolic link from one to the other *original*.

If an existing symbolic link is replaced, the replacement is created with the identical owner and group, as the existing link was. If a new symbolic link is created, the owner and group of the new symbolic link is set as the item it is pointing to. A new standalone symbolic link, or if the user using the **symlink** command is not allowed to set the user and group memberships, the new symbolic link is created with the owner-ships of the creating user.

OPTIONS

-h

usage message.

-l

link must be an existing symbolic link. If it isn't an existing symbolic link, it will not be removed.

original

original file.

link

symbolic link.

ENVIRONMENT

-

EXIT STATUS

| | |
|----------|---|
| 0 | no error. |
| 1 | cannot remove existing symlink. |
| 2 | <i>link</i> does not exist or is not a symlink. |
| 3 | cannot create symlink. |
| 4 | usage printed. |

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ln(1)**

NOTES

-

BUGS

-

AUTHOR

symlink was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sys – start edrc with a special WA2L/edrc maintenance configuration

SYNOPSIS

edrc/bin/sys [**-h** | **-V**]

sys [**-s**] [**-t**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

Short start of **edrc** with an other configuration which points to an own script tree. Internally **sys** calls **edrc -c edrc.sys.cfg -n sys_@ID@** .

This configuration is part of the WA2L/edrc package and the menu should not be modified, due to the fact that updates are provided thru patches and all changes would be lost or the changes made would question the correct operation of the script tree.

The menu points provide an user interface to some installation, patching and package distribution tasks.

The session name (as shown in the menu) is automatically set to **sys_<id>** . Where **<id>** is the process id of the started **edrc** instance if not already a session with the same name exists, if so the **<id>** is set to a random number.

If additional short starts are needed, create a symlink from the new short start command to **sat** (see section **EXAMPLES**).

OPTIONS

- h** usage message.
- V** print version and patch level of **edrc**. For an explanation of the release numbering system see **edrcrevision(1)**.
- s** silent startup. Startup without showing the EDRC banner.
- t** no terminal initialization. Sometimes the terminal initialization causes unwanted behavior. To skip terminal initialization, invoke **edrc** with this option.

ENVIRONMENT

-

EXIT STATUS

see **edrc**(1m).

FILES

etc/edrc.sys.cfg

configuration file of **sys**, see **edrc.cfg**(4) for more information.

Other files see section **FILES** in **edrc**(1m).

EXAMPLES**1) create a new short start**

This creates a new short start command **new_shortstart** that will load the configuration file **edrc.new_shortstart.cfg**. The session name will automatically be set to **NEW_SHORTSTART_<id>** .

```
[ /root ]
[ root@rh7mzv7t001 ] [bash]: cd ~edrc/bin

[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ] [bash]: ln -s sat new_shortstart
```

SEE ALSO

sat(1), **edrc**(1m), **edrc.cfg**(4), **edrcintro**(1), **edrcrevision**(1), **edrcsetup**(1)

NOTES

-

BUGS

-

AUTHOR

sys was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sysconfig – configure a Unix/Linux system

SYNOPSIS

edrc/bin/sysconfig [**-h**]

sysconfig **-s** *src_dir* [**-r** *root_dir*] [**-b** *backup_dir*] [**-o** *OSID*] [**-f**]

sysconfig **-p** [**-r** *root_dir*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

With **sysconfig** you can configure a UNIX or Linux system with a single command using a set of prepared configuration files.

In addition to the prepared files, pre- and post exec scripts enable to run commands prior and after the application of the configuration files. Therefore the operating system configuration can be automated completely.

sysconfig should be used to configure the vanilla operating system as delivered by the vendor instead of including the configuration to the image as installed by Sun JumpStart, Hewlett-Packard Ignite, RedHat Kickstart, SuSE AutoYast etc. .

This has the advantage that the set of changed configurations is well known and it is more efficient to configure a future operating system version with the equal quality. Looking at the configuration repository it is easy to evaluate which files were adjusted to configure the operating system, without relying on configuration documentation. When including the configuration in the related image as mentioned above it is harder to see which files were changed and the possibility of not documenting the configuration changes is high. Furthermore it is more efficient to change the configuration in the **sysconfig** repository due to the fact that the repository has the same structure for all operating systems (HP-UX, Solaris, Linux, ...) and the user does not need to know the different image formats / install methods provided by the operating system vendor.

FEATURES

- Currently supported operating systems: Solaris, HP-UX, Linux
- Pre- and post exec support
- Backup of all removed/changed files
- Protection of files that change after initial system configuration
- Simple repository structure, identical to the target operating system directory structure
- Repository structure is identical for all supported operating systems
- Possibility of applying multiple configurations to a system
- Tools to maintain the configuration repository: **sysvi**(1), **syscp**(1), **sysdiff**(1)
- Tools to use in pre- and post exec scripts: **apply2file**(1), **is_config_byhand**(3)

CONFIGURATION STEPS

The following steps are performed during a system configuration. Some of them are optional.

- 1) execute the **check** script if it exists. This script is used to ensure that a certain configuration is only applied to a system if the criteria implemented in the **check** script are met. If this script returns 0 or if this script does not exist it is allowed to configure the system and the following steps are executed.
- 2) write all settings to the sysconfig-settings file. This file will later be read if you invoke **sysconfig -p** and **svi** and is therefore used to identify which configuration(s) has been applied to a system.
- 3) execute the **pre_exec** script if it exists.
- 4) backup files listed in the **backup** file.
- 5) remove files:
 - a) collect information about files protected by **config_byhand** settings applied to the system.
 - b) backup files listed in the **remove** file and not listed in any applied **config_byhand** file.
 - c) remove files listed in the **remove** file and not listed in any applied **config_byhand** file.
- 6) install files from the source root dir:
 - a) check which files has to be installed (only files which differ from the system or have different permissions or ownership and are not listed in any applied **config_byhand** file will be installed)
 - b) backup files which will be overwritten in the install process in step 6c).
 - c) install files evaluated in step 6a).
- 7) execute the **post_exec** script if it exists.
- 8) write **config_byhand** to **<root_dir>/EDRC.config_byhand<.name>** if it exists and it is not protected by an already applied **config_byhand** file.
If the **CONFIG_BYHAND** setting in the **sysconfig.cfg** file is set to **MERGE** the new entries and the already existing entries in the applied **config_byhand** file are merged and written to the **<root_dir>/EDRC.config_byhand<.name>** file, else the applied settings are replaced with the new entries.
- 9) compress **.cpio** file that contains the backedup files to a **.cpio.gz** file.

OPTIONS

- h** usage message.

- s** *source_directory*
path to the directory which contains the filestructure used by **sysconfig** to configure the system.

- p** print the current active configuration(s) of a system. It is possible that a system has multiple configurations active for certain directories. This is especially the case if you use named configurations or if you configure diskless clients on a server.

- r** *root_dir*
root directory where you apply or query the configuration. You can apply all configurations to any target *root directory* only if you write proper **pre_exec** and **post_exec** scripts and if you list only relative path names within the **backup** and **remove** files explained in section **SYSCONFIG-DATASTRUCTURE**. The default for *root_dir* is */*.

- b** *backup_dir*
directory where to save files backed up from the system ahead of overwriting. The default can be changed in **sysconfig.cfg**, see **sysconfig.cfg(4)** for more information.

- f** force configuration. If you specify this option all **config_byhand** settings are ignored.

- o** *OSID* in normal cases you don't have to set the operating system id (*OSID*) due to the fact that it will be resolved by **sysconfig** using the **osid** command internally. But for testing purposes or if you have to configure diskless clients with different operating system ids than the server, it might be required to set a specific *OSID*. See **osid(3)** for more information.

SYSCONFIG-DATASTRUCTURE

This directory- and filestructure is needed by **sysconfig**. Some files are optional.

<source_directory>

this directory contains the configuration directory structure and all definitions needed by **sysconfig**. This directory has to be specified in the **-s** *src_dir* option of **sysconfig**.

Optional: NO

Example: development.acme.com

Default: -

<source_directory>/<OSID>

operating system ID evaluated by **osid** or set with the **-o** option. If you have to configure different operating systems within an environment you have simply to create an additional *OSID* directory in the same *source_directory*. See **osid(3)** for more information.

Optional: NO

Example: HP-11

Default: -

<source_directory>/<OSID>/README

free-text description and notes to the configuration. This file is to document specialties of the configuration and is not computed by **sysconfig**.

Optional: YES

Example:

```
#
# README - Free text explanations/notes to the configuration
#
# [00] 02.05.2008 CWa    Initial Version
#
```

Default: -

<source_directory>/<OSID>/check

execute the check script if it exists. This script is used to ensure that a certain configuration is only applied to a system if the criteria implemented in the check script are met. If this script returns 0 it is allowed to configure the system and the following steps are executed. See also the description of the tool directory **var/sysconfig/tools/** to see how to save time and effort in writing check scripts.

Optional: YES

Example:

```
#!/bin/sh
#
# check - check if configuration is allowed
#
# [00] 08.02.2003 CWa    Initial Version
#
#
test "$DEBUG" = True && set -x

Targethost=acme054

if [ "`hostname`" = "$Targethost" ]; then
    exit 0
else
    echo "apply this config on $Targethost only!"
    exit 1
fi
```

Default: allow configuration (exitcode=0)

<source_directory>/<OSID>/description

this file contains a single line with a description of the configuration. This description is displayed if you invoke **sysconfig -p** on a configured system.

Optional: YES

Example: ACME Development Department Configuration

Default: no description defined

<source_directory>/<OSID>/name

name of a configuration. This gives you the possibility to apply multiple configurations to the same *root_dir*. This is useful if you have to configure cluster packages (e.g. MC/Service-Guard) which share the *root_dir* with the operating system configuration. Normally you will not name the operating system configuration, but you will name special configurations (such as cluster packages).

Optional: YES

Example: BO

Default: -

<source_directory>/<OSID>/pre_exec

execute pre configuration tasks which cannot be handled with the other mechanisms. A common task in a pre exec script could be: stopping daemon which should not run during configuration. See also in section **ENVIRONMENT** for additional important hints for writing post and pre exec scripts. If you modify files on the system you should use **is_config_byhand** to ensure not to overwrite a protected file. See also the description of the tool directory **var/sysconfig/tools/** to see how to save time and effort in writing pre- and post exec scripts.

Optional: YES

Example:

```
#!/bin/sh
#
# pre_exec - pre exec script
#
# [00] 04.04.2003 CWa    Initial Version
#
#

test "$DEBUG" = True && set -x

touch $SYSCONFIG_ROOTDIR/exAppl/connect.lck
/sbin/init.d/nfs.client stop
```

Default: -

<source_directory>/<OSID>/backup

a list of files to be backuped in step 4). Do not use absolute pathnames, use only relative ones. This ensures that your configuration can be applied to any *root_dir* which is important if you plan to configure diskless clients on a server or simply for testing your configuration. Lines starting with a # or empty lines are considered as comments.

Optional: YES

Example:

```
#
# backup - list of files to be backuped
#
# [00] 08.02.2003 CWa    Initial Version
#
etc/passwd
```

Default: no extra file backup

<source_directory>/<OSID>/config_byhand

a list of files to be protected from overwriting and removal after the initial run of **sysconfig**.

Do not use absolute pathnames, use only relative ones. This ensures that your configuration can be applied to any *root_dir* which is important if you plan to configure diskless clients on a server or simply for testing your configuration. Lines starting with a # or empty lines are considered as comments.

This file is copied to **<root_dir>/EDRC.config_byhand<.name>**. If you change a file on the system and you think it might be handled by **sysconfig** you should add the filename to **.EDRC.config_byhand<.name>** to protect it from overwriting. This is also true for the **.EDRC.config_byhand<.name>** itself.

To dynamically merge the settings defined in an applied **<root_dir>/EDRC.config_byhand<.name>** file and the settings of a **config_byhand** file in a **sysconfig** repository to be applied, set the **CONFIG_BYHAND** setting in the **sysconfig.cfg** configuration file to **MERGE**. In this case it has to be ensured that the **.EDRC.config_byhand<.name>** is not listed in the **.EDRC.config_byhand<.name>** or the **config_byhand** file, because this would protect the file from further change.

post_exec and **pre_exec** scripts should use **is_config_byhand** if they modify files to ensure it is not protected by a **config_byhand** setting.

Optional: YES

Example:

```
#
# config_byhand - list of config by hand files
#
```

```
# [00] 26.10.2003 CWa    Initial Version
#
etc/passwd
etc/dfs/dfstab
var/spool/cron/crontabs/root
```

Default: no files to be configured by hand

<source_directory>/<OSID>/remove

a list of files to be removed in step 5). Do not use absolute pathnames, use only relative ones. This ensures that your configuration can be applied to any *root_dir* which is important if you plan to configure diskless clients on a server or simply for testing your configuration. Lines starting with a # or empty lines are considered as comments.

Optional: YES

Example:

```
#
# remove - list of files to be removed
#
# [00] 08.02.2003 CWa    Initial Version
#
etc/opt/app/lock.lk
```

Default: no extra file backup

<source_directory>/<OSID>/root/

in this directory the system configuration files resist. All files in this directory are copied to the system during the configuration if the file differs from the file already present on the target system. Symlinks are copied always.

Optional: NO

<source_directory>/<OSID>/post_exec

execute post configuration tasks which cannot be handled with the other mechanisms. A common task in a post exec script could be: setting of the default router entry or starting daemon which should not run during configuration. See also in section **ENVIRONMENT** for additional important hints for writing post and pre exec scripts. If you modify files on the system you should use **is_config_byhand** to ensure not to overwrite a protected file. See also the description of the tool directory **var/sysconfig/tools/** to see how to save time and effort in writing post- and pre exec scripts.

Optional: YES

Example:

```
#!/bin/sh
#
# post_exec - post exec script
#
# [00] 04.04.2003 CWa Initial Version
#
#
test "$DEBUG" = True && set -x

rm $SYSCONFIG_ROOTDIR/exAppl/connect.lck
/sbin/init.d/nfs.client start
```

Default: -

ENVIRONMENT

The following environment variables are exported to the **pre_exec** and **post_exec** scripts. It is highly recommended to use those variables. Otherwise you cannot use all features of **sysconfig**.

\$SYSCONFIG_ROOTDIR

this variable equals to the value specified in the **-r** option. Your **pre_exec**, **post_exec** and **check** scripts will only work for every target directory if you use this variable wherever you access a file on the system.

\$SYSCONFIG_FORCE

this variable is set to 'True' if **sysconfig** is called with the **-f** option. If you use **is_config_byhand** it will return 'False' for each file in this case.

EXIT STATUS

- 0** no error.
- 1** you started **edrc** using the wrong operating system user.
- 2** operating system is not supported. See **osid(3)** if you get this error.
- 4** usage listed.
- 6** the configuration file does not exist.

FILES

etc/sysconfig.cfg

configuration file of **edrc**, see **sysconfig.cfg(4)** for more information.

var/log/sysconfig.log

logfile of **sysconfig**.

var/sysconfig/tools/<OSID>

additional tools (scripts) that can be called in the **check**, **pre_exec** and **post_exec** scripts. If a configuration supports multiple operating systems, what is normally the case in heterogeneous environments, you can place operating system dependent scripts in the related directory. If you like to set the default router entry on your Linux, HP-UX and Solaris systems (this is very different on those systems) in the **post_exec** script you can create a script called for instance **'set_default_router'** for each operating system and place it into the related directory. This enables you to save effort in writing the **post_exec** script for all the operating systems you like to configure and simplifies those scripts because you don't have to include the whole code into multiple post exec scripts. If multiple operating systems can be handled identical, a script can be placed into the **var/sysconfig/tools/default** folder and there is no need to have to place identical copies into more then one **<OSID>** directory. The search order is to first look for the tool script in the operating system related directory (**<OSID>**) and then in the **default** directory. You should (of course) synchronize the tool directory contents over all your WA2L/edrc installations to fully benefit from this feature. See **filedist(1)** for instructions how to distribute files easily.

See also: **sysconfig.tools(3)**.

var/sysconfig/tools/default

see description of directory **var/sysconfig/tools/<OSID>**.

See also: **sysconfig.tools(3)**.

var/sysconfig/development.acme.com

example and template **SYSCONFIG-DATASTRUCTURE** tree. This tree can be used as a template and copied to your specific configuration.

<source_directory>

this directory contains the configuration definitions. See section **SYSCONFIG-DATASTRUCTURE** for an explanation of every file within this directory.

<source_directory>/sysconfig.cfg

configuration file of **edrc**, see **sysconfig.cfg(4)** for more information. This file overrides settings made in **etc/sysconfig.cfg**.

<root_dir>/EDRC.sysconfig <root_dir>/EDRC.sysconfig.<name>

settings file which contains information about current active configuration. This file is read by **sysconfig -p** and **svi**.

<root_dir>/EDRC.config_byhand <root_dir>/EDRC.config_byhand.<name>

a list of files to be protected from overwriting after the initial run of **sysconfig**.

<root_dir>/config_byhand

a list of files to be protected from overwriting after the initial run of **rmtconfig** a system configuration tool provided by the SFI-Director. New entries should not be entered into this file, it is only considered to protect systems configured with **rmtconfig** from miss-configuration.

EXAMPLES

0) naming of the system configuration sources

There is no restriction of the naming of the configuration sources (**CONFIGURATION**).

However, it proved to make sense to use a naming scheme as known from the Domain Name Service (DNS) due to the fact that it is well known and the hierarchical structure that is expandable and allows to start with a very simple configuration setup and enables to specify also very specialized configurations with clear naming.

Configuration naming convention following the DNS example:

| CONFIGURATION | NAME | Explanation |
|-----------------------|------|--|
| acme.ch | base | The base configuration defines the system behaviour for the whole ACME enterprise. This configuration holds all files that are common in all Unix environments in the ACME corporation. All environment dependent configurations are done applying the related configuration on top of this one. |
| test.acme.ch | | This configuration holds only the files that are special to the TEST environment. For a complete system configuration, first the 'acme.ch' configuration and then the 'test.acme.ch' configuration has to be applied. |
| production.acme.ch | | This configuration holds only the files that are special to the PRODUCTION environment. |
| preproduction.acme.ch | | This configuration holds only the files that are special to the PREPRODUCTION environment. |

See also the following examples.

1) apply a system configuration to a vanilla system

This is an output example of the use of **sysconfig** on a vanilla HP-UX (OSID = HP-11) system. This configuration has no **pre_exec** script, but has a **post_exec** script. At this stage no **config_byhand** information is available. See also next example, where the configuration is re-applied.

If the hostname is not set, set it prior to applying the configuration using the **hostname(1)** command:

```
[ /root ]
[ root@unknown ][-sh]: hostname acme001
```

Apply the configuration:

```
[ /root ]
```

```
[ root@acme001 ][-sh]: ~edrc/bin/sysconfig \
-s /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch
```

sysconfig - configure a Unix/Linux system, by Chr. Walther

```
DATE:          Mon Jun  4 16:00:05 METDST 2007
HOSTNAME:      acme001
CONFIGURATION: acme.ch/HP-11
ROOTDIR:       /
BACKUPFILE:    2007-06-04_16.00.05__sysconfig.cpio.gz
FORCE:         False
```

```
configure system acme001 ...
  execute check script ...
    allow configuration (exitcode=0)
done.
write settings ...
  /.EDRC.sysconfig.base written
done.
execute pre exec script ...
done.
backup files in / ...
  evaluate files to backup ...
    etc/issue
done.
  backup current files ...
    etc/issue
    1 block
done.
done.
remove files in / ...
  collect config_byhand settings ...
    <no files>
done.
  evaluate files to remove ...
    <no files>
done.
  backup current files ...
    <no files to backup>
done.
  remove files ...
    <no files to remove>
done.
done.
install files and directories in / ...
  collect config_byhand settings ...
    <no files>
done.
  evaluate files to install ...
    .secure
    .secure/etc
    anw
    apl
    dat
```

```
etc
etc/auto_apl
etc/auto_direct
etc/auto_home
:
:
etc/default
etc/default/security
etc/default/tz
:
:
etc/group.description
etc/hosts
etc/hosts.equiv
etc/inetd.conf
etc/issue
etc/lvmvg
etc/mail
etc/mail/aliases
etc/mail/mailcap
etc/mail/sendmail.cf
etc/mail/sendmail.cw
etc/mail/service.switch
etc/motd
etc/netgroup
etc/nsswitch.conf
etc/ntp.conf
etc/opt
:
:
etc/rc.config.d
etc/rc.config.d/auditing
etc/rc.config.d/cmcluster
etc/rc.config.d/nddconf
etc/rc.config.d/netconf
etc/rc.config.d/netdaemons
etc/rc.config.d/nfsconf
etc/rc.config.d/samba
etc/rc.config.d/xntpd
etc/resolv.conf
etc/securetty
etc/services
:
:
opt/iexpress/sudo
opt/iexpress/sudo/etc
opt/iexpress/sudo/etc/sudoers
opt/iexpress/sudo/etc/sudoers.main
opt/iexpress/sudo/etc/sudoers.user
opt/patrol
opt/ssh
opt/ssh/etc
opt/ssh/etc/ssh_config
opt/ssh/etc/sshd_config
```



```
root
root/.bashrc
root/.cshrc
:
:
var/spool
var/spool/cron
var/spool/cron/crontabs
var/spool/cron/crontabs/root
done.
backup current files ...
etc/auto_master
etc/default/tz
etc/group
etc/hosts
etc/inetd.conf
etc/issue
:
:
opt/ssh/etc/sshd_config
var/adm/cron/at.allow
var/adm/cron/cron.allow
var/opt/perf/parm
var/spool/cron/crontabs/root
373 blocks
done.
install new files ...
/.secure
/.secure/etc
/anw
/apl
/dat
/etc
/etc/auto_apl
/etc/auto_direct
:
:
/var/opt/perf
/var/opt/perf/parm
/var/spool
/var/spool/cron
/var/spool/cron/crontabs
/var/spool/cron/crontabs/root
8367 blocks
done.
done.
execute post exec script ...
  apply2file - install a data stream as file, by Chr. Walther

process file '//etc/issue' ...
  data written to file
  permissions of original file preserved
done.
post_exec-INFO: remove '//net' directory
```

```

apply2file - install a data stream as file, by Chr. Walther

process file '//etc/rc.config.d/netconf' ...
    data written to file
    permissions of original file preserved
done.
apply2file - install a data stream as file, by Chr. Walther

process file '//.secure/etc/audnames' ...
    data written to file
    permissions set to (root:sys 600)
done.
done.
write config_byhand ...
    /.EDRC.config_byhand.base written
done.
done.

```

2) re-apply a system configuration to a system already configured with sysconfig

This is the re-applying of the same configuration as applied in the example 1) above. See the now expanded list of files protected by **config_byhand**. In this example all files residing on the system are still identical with the files in the repository, therefore it was not needed to re-apply any file to the system.

```

[ /root ]
[ root@acme001 ][-sh]: ~edrc/bin/sysconfig \
    -s /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch

sysconfig - configure a Unix/Linux system, by Chr. Walther

DATE:           Mon Jun  4 16:13:27 METDST 2007
HOSTNAME:       acme001
CONFIGURATION:  acme.ch/HP-11
ROOTDIR:        /
BACKUPFILE:     2007-06-04_16.13.27__sysconfig.cpio.gz
FORCE:          False

configure system acme001 ...
    execute check script ...
        allow configuration (exitcode=0)
done.
write settings ...
    /.EDRC.sysconfig.base written
done.
execute pre exec script ...
done.
backup files in / ...
    evaluate files to backup ...
        etc/issue
done.
backup current files ...
    etc/issue

```

```

        1 block
    done.
done.
remove files in / ...
    collect config_byhand settings ...
        .EDRC.config_byhand.base
        .secure/etc/audnames
        etc/auto_dat
        etc/auto_master
        etc/ftpd/ftpusers
        etc/group
        etc/group.description
        :
        :
        etc/passwd.ownermap
        etc/rc.config.d/cmcluster
        etc/rc.config.d/netconf
        etc/su.allow
        opt/ACME/data/input/UsersToCheck
        root/.mycshrc
        root/.mykshrc
        root/.mylogin
        root/.myprofile
        tcb/files/auth/system/default
        var/spool/cron/crontabs/root
    done.
evaluate files to remove ...
    <no files>
done.
backup current files ...
    <no files to backup>
done.
remove files ...
    <no files to remove>
done.
done.
install files and directories in / ...
    collect config_byhand settings ...
        .EDRC.config_byhand.base
        .secure/etc/audnames
        etc/auto_dat
        etc/auto_master
        etc/ftpd/ftpusers
        etc/group
        etc/group.description
        :
        :
        etc/passwd.ownermap
        etc/rc.config.d/cmcluster
        etc/rc.config.d/netconf
        etc/su.allow
        opt/ACME/data/input/UsersToCheck
        root/.mycshrc
        root/.mykshrc

```

```

    root/.mylogin
    root/.myprofile
    tcb/files/auth/system/default
    var/spool/cron/crontabs/root
done.
evaluate files to install ...
    <no files>
done.
backup current files ...
    <no files to backup>
done.
install new files ...
    0 blocks
done.
done.
execute post exec script ...
    apply2file - install a data stream as file, by Chr. Walther

    process file '//etc/issue' ...
        data is identical with existing file, aborting
done.
post_exec-INFO: remove '//net' directory
    apply2file - install a data stream as file, by Chr. Walther

    process file '//etc/rc.config.d/netconf' ...
        data is identical with existing file, aborting
done.
    apply2file - install a data stream as file, by Chr. Walther

    process file '//.secure/etc/audnames' ...
        data is identical with existing file, aborting
done.
done.
write config_byhand ...
    /.EDRC.config_byhand.base protected, not written
done.
done.

```

3) print applied configuration

Print the configuration applied to a system using **sysconfig**. This system has one configuration applied:

```

[ / ]
[ root@acme001 ][-sh]: ~edrc/bin/sysconfig -p

sysconfig - configure a Unix/Linux system, by Chr. Walther

/ -

CONFIG:      acme.ch
NAME:
OSID:        HP-11

```

```

SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch
ROOTDIR:     /
BACKUPDIR:   /opt/edrc/var/backup
BACKUPFILE:  2007-06-04_16.13.27__sysconfig.cpio.gz
DESCRIPTION: BASE PRODUCTION SERVER CONFIGURATION for Operating System
EXECUTED:    Mon Jun  4 16:13:27 METDST 2007

```

4) system having multiple configurations applied

This is an efficient usage of **sysconfig** using two configurations.

The base configuration (acme.ch) defines the system behaviour for the whole ACME enterprise and the unnamed environment dependent configuration (production.acme.ch) defines the special configuration for the servers in the PRODUCTION environment.

```

[ /root ]
[ root@acme002 ][-sh]: ~edrc/bin/sysconfig -p

```

sysconfig - configure a Unix/Linux system, by Chr. Walther

/ -

```

CONFIG:      production.acme.ch
NAME:
OSID:        HP-11
SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch
ROOTDIR:     /
BACKUPDIR:   /opt/edrc/var/backup
BACKUPFILE:  2005-07-12_20.09.27__sysconfig.cpio.gz
DESCRIPTION: PRODUCTION SERVER CONFIGURATION for Operating System
EXECUTED:    Tue Jul 12 20:09:27 METDST 2005

```

/ - base

```

CONFIG:      acme.ch
NAME:        base
OSID:        HP-11
SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch
ROOTDIR:     /
BACKUPDIR:   /opt/edrc/var/backup
BACKUPFILE:  2004-11-23_05.28.15__sysconfig.cpio.gz
DESCRIPTION: BASE SERVER CONFIGURATION for Operating System
EXECUTED:    Tue Nov 23 05:28:15 EST 2004

```

5) system having multiple configurations plus cluster support

This is a more advanced usage of **sysconfig** also including the configuration of cluster packages.

This output is from a system used in normal operations for PREPRODUCTION and in a

disaster case as **PRODUCTION**. The system will be reconfigured (also using **sysconfig**) during disaster recovery to give it the **PRODUCTION** configuration.

This system has five configurations applied. In this case the *base* (acme.ch) configuration contains all basic configuration files of the ACME enterprise. The *preproduction* (preproduction.acme.ch) configuration is applied on top of the *base* configuration and contains all basic configuration of the **PREPRODUCTION** environment that will not be changed during a disaster recovery. The unnamed (normal.preproduction.acme.ch) configuration is applied on top of the *preproduction* configuration and contains only the files that are needed to configure the system as a **PREPRODUCTION** system in **NORMAL OPERATIONS**. This part of the configuration will be replaced with the unnamed (disaster.preproduction.acme.ch) configuration to reconfigure the system in case of disaster and give it the **PRODUCTION** identity.

The configurations *dwh_db1* and *dwh_db2* are the configurations of cluster packages, that are reconfigured during disaster recovery, too.

To ensure that the cluster package configuration is only displayed on the node where the cluster package is running, prior to the execution of **sysconfig** a symbolic link to the settings file has to be created. This will place the settings file to a moving file system:

```
[ / ]
[ root@acme003 ][-sh]: ln -s /data_dwh1/.EDRC.sysconfig.dwh_db1 \
                        .EDRC.sysconfig.dwh_db1
```

The symbolic link has to be created on each node where the cluster package is able to start.

In this case the filesystems **/data_dwh1/**, **/ora0[1-3]_dwh1/** are moved with the cluster package, that's also the reason why the configurations of the cluster packages are needed to be applied to the root directory (**/**).

```
[ /root ]
[ root@acme003 ][-sh]: ~edrc/bin/sysconfig -p
```

sysconfig - configure a Unix/Linux system, by Chr. Walther

/ -

```
CONFIG:      normal.preproduction.acme.ch
NAME:        normal.preproduction.acme.ch
OSID:        HP-11
SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/normal.preproductio
ROOTDIR:     /
BACKUPDIR:   /opt/edrc/var/backup
BACKUPFILE:  2005-07-12_20.30.04__sysconfig.cpio.gz
DESCRIPTION: NORMAL OPERATION PREPRODUCTION SERVER CONFIGURATION for
EXECUTED:    Tue Jul 12 20:30:04 METDST 2005
```

/ - preproduction

```
CONFIG:      preproduction.acme.ch
NAME:        preproduction
OSID:        HP-11
SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/preproduction.acme.
ROOTDIR:     /
```

```

BACKUPDIR:    /opt/edrc/var/backup
BACKUPFILE:   2004-11-25_14.54.06__sysconfig.cpio.gz
DESCRIPTION:  BASE PREPRODUCTION SERVER CONFIGURATION for Operating Sy
EXECUTED:     Thu Nov 25 14:54:06 MET 2004

```

/ - base

```

CONFIG:       acme.ch
NAME:         base
OSID:         HP-11
SOURCE:       /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch
ROOTDIR:      /
BACKUPDIR:    /opt/edrc/var/backup
BACKUPFILE:   2004-11-24_15.00.01__sysconfig.cpio.gz
DESCRIPTION:  BASE SERVER CONFIGURATION for Operating System
EXECUTED:     Wed Nov 24 15:00:01 MET 2004

```

/ - dwh_db1

```

CONFIG:       dwh_db1.normal.preproduction.acme.ch
NAME:         dwh_db1
OSID:         HP-11
SOURCE:       /opt/edrc/scripts/ACME-sat/sysconfig/dwh_db1.normal.prep
ROOTDIR:      /
BACKUPDIR:    /opt/edrc/var/backup
BACKUPFILE:   2007-08-20_12.38.37__sysconfig.cpio.gz
DESCRIPTION:  NORMAL OPERATION PREPRODUCTION CONFIGURATION for Cluster
EXECUTED:     Mon Aug 20 12:38:38 METDST 2007

```

/ - dwh_db2

```

CONFIG:       dwh_db2.normal.preproduction.acme.ch
NAME:         dwh_db2
OSID:         HP-11
SOURCE:       /opt/edrc/scripts/ACME-sat/sysconfig/dwh_db2.normal.prep
ROOTDIR:      /
BACKUPDIR:    /opt/edrc/var/backup
BACKUPFILE:   2007-08-20_12.38.42__sysconfig.cpio.gz
DESCRIPTION:  NORMAL OPERATION PREPRODUCTION CONFIGURATION for Cluster
EXECUTED:     Mon Aug 20 12:38:43 METDST 2007

```

SEE ALSO

edrcintro(1), **apply2file(1)**, **filedist(1)**, **filelink(1)**, **is_config_byhand(3)**, **ksh(1)**, **lscomp(1)**, **llcomp(1)**, **osid(3)**, **sh(1)**, **svi(1)**, **sysconfig.cfg(4)**, **sysconfig.tools(3)**, **syscp(1)**, **sysdiff(1)**, **sysvi(1)**

NOTES

The implementation of the **config_byhand** feature as known from the SFI-Director **rmtconfig** command.

BUGS

-

AUTHOR

sysconfig was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sysconfig.cfg – configuration file for sysconfig

SYNOPSIS

edrc/etc/sysconfig.cfg
<src_dir>/sysconfig.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **sysconfig** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**START_USER**

This is the user **edrc** has to be started with.

Example: **START_USER=eroot**

Default: **START_USER=root**

BACKUP_DIR

sysconfig saves all files ahead of removing or installing the new files scripts to this directory.

Example: **BACKUP_DIR=/var/opt/edrc/backup**

Default: **BACKUP_DIR=var/backup**

LOG

Log output dir of **sysconfig**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: **LOG=/var/opt/edrc/log**

Default: LOG=var/log

CONFIG_BYHAND

With this setting the behaviour of the applying of the **config_byhand** settings can be configured.

When it is set to *REPLACE* the already applied **config_byhand** file is replaced by the **config_byhand** file of the **sysconfig** repository to be applied.

If it is set to *MERGE*, the definitions of the already applied **config_byhand** file and the contents of the **config_byhand** file in the **sysconfig** repository is merged together.

Example: CONFIG_BYHAND=MERGE

Default: CONFIG_BYHAND=REPLACE

SEE ALSO

sysconfig(1), **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

sysconfig.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sysconfig.tools – list of all sysconfig tools distributed with WA2L/edrc

SYNOPSIS

-

AVAILABILITY

WA2L/edrc

DESCRIPTION

This man page lists all additional system configuration tools (scripts) that can be called in the **check**, **pre_exec** and **post_exec** scripts that are part of the core WA2L/edrc package and distributed with it.

If you write an own system configuration tool do not list it in this man page, due to the fact that your change would be lost after an upgrade of WA2L/edrc. If you send your tool to the author of WA2L/edrc and it is decided to include it into the core distribution, your command will be added to this list.

If a man page is written for a system configuration tool, the manpage name convention is **sysconfig.tools.my_command.3** and it has to be placed into the man page directory **man/all/man3**.

If a configuration supports multiple operating systems, what is normally the case in heterogeneous environments, you can place operating system dependent scripts into the directory **var/sysconfig/tools/<OSID>**, where **<OSID>** is the operating system id as returned by the **osid(1)** command, instead of coding everything directly into the **check**, **pre_exec** or **post_exec** script.

E.g.: if you like to set the default router entry on your Linux, HP-UX and Solaris systems (this is very different on those systems) in the **post_exec** script you can create a script called for instance **'set_default_router'** for each operating system and place it into the related directory. This enables you to save effort in writing the **post_exec** script for all the operating systems to be configured and simplifies those scripts because you don't have to include the whole code into multiple post exec scripts.

If multiple operating systems can be handled identical, a script can be placed into the **var/sysconfig/tools/default** folder and there is no need to have to place identical copies into more then one **<OSID>** directory.

The search order is to first look for the tool script in the operating system related directory (**<OSID>**) and then in the **default** directory. You should (of course) synchronize the tool directory contents over all your WA2L/edrc installations to fully benefit from this feature. See **filedist(1)** for instructions how to distribute files easily.

SYSCONFIG TOOLS OF CORE WA2L/edrc

default/create_users

SYNOPSIS

```
cat <<EOM | create_users [ passwd_file ]
tools-acme:*:901:7000:ACME Users Tools, Software, 2008-05-31 2008-05-
tools-all:*:902:7000:all Users Tools, Software, 2008-05-31 2008-05-31
ACME:*:904:7000:ACME Scripts, Software, 2008-05-31 2008-05-31:/opt/AC
EOM
```

DESCRIPTION

create user(s) on a system based on a **/etc/passwd** format input provided via **stdin**. A user created using **create_users** is first removed from the system and then added again to ensure the settings are correct. The user is removed with the system's native **userdel (8)** and added with **useradd(8)**, but if one of the commands fails, the related step is performed by modifying the **/etc/passwd** file directly.

SEE ALSO

sysconfig.tools.create_users(3)

NOTES

-

AUTHOR

create_users was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

{HP-11|HP-11i|HP-11ia}/edit_netconf

SYNOPSIS

```
edit_netconf netconf_file | ...
```

DESCRIPTION

interactively edit a file having the structure of the **/etc/rc.config.d/netconf** file as known on HP-UX.

In addition to the standard network interface fields (**INTERFACE_NAME**, **IP_ADDRESS**, **SUBNET_MASK**, **BROADCAST_ADDRESS**, **INTERFACE_MODULES**, **INTERFACE_STATE**, **DHCP_ENABLE**) the field **INTERFACE_DESCRIPTION** is computed to define a textual description of the purpose for each interface.

In addition to the standard routing definition fields (**ROUTE_DESTINATION**, **ROUTE_GATEWAY**, **ROUTE_COUNT**, **ROUTE_ARGS**) the field **ROUTE_DESCRIPTION** is computed to define a textual description of the purpose for each route entry.

The result of the editing is sent to **stdout** to allow further data processing.

SEE ALSO

sysconfig.tools.edit_netconf(3)

NOTES

the OS-ids **HP-11i** and **HP-11ia** are symbolic links to **HP-11** that's why the **edit_netconf** is not located in the **default** directory.

AUTHOR

edit_netconf was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

ENVIRONMENT

See **sysconfig(1)** section **ENVIRONMENT**.

EXIT STATUS

-

FILES

var/sysconfig/tools/<OSID>
operating system dependent system configuration tools.

var/sysconfig/tools/default
non operating system dependent system configuration tools or tools that can be used on multiple operating systems.

EXAMPLES

-

SEE ALSO

edrcintro(1), **sysconfig(1)**, **sysconfig.tools.edit_netconf(3)**, **sysconfig.tools.create_users(3)**, **sysconfig.tools.<sysconfig_tool>(3)**

NOTES

-

BUGS

-

AUTHOR

sysconfig.tools was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sysinfo – print information on overall system statistics

SYNOPSIS

edrc/bin/sysinfo [**-h** | **-a**]

sysinfo [**-u** | **-l** | **-r** | **-s** | **-n**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

With the **sysinfo** command information about the overall system statistics is printed.

If called with the **-a** option, all overall statistics information is provided in a plain human readable format.

However, the main purpose of the **sysinfo** command is to print out dedicated information with the options **-u**, **-l**, **-r**, **-s** or **-n**. The output of those options is formatted in a fashion that it is possible to compute it easily with other commands.

Especially the output of the **uptime(1)** command provided by the operating system is often very hard to use for further data processing, due to the fact that the output changes completely depending on the point in time when called. Therefore the output of **sysinfo -u** is much easier to process.

The format of all outputs (except **-a**) is:

<date>;<seconds since Epoch>{;<value>}

Example (**sysinfo -u**):

2006-09-30 00:04:50;1159567490;349541

OPTIONS

- h** usage message.
- a** all overall system statistics information in a plain human readable format. This output is not intended for further data processing.
- u** uptime (=time since last boot) in seconds [one value field].
- l** 1, 5 and 15 minute load average [three value fields].
- r** total usable main memory, total available memory, amount of shared memory and memory used by buffers in bytes [four value fields].

- s** total swap space and swap space still available in bytes [two value fields].
- n** number of current processes [one value field].

EXIT STATUS

- 0** no error.
- 1** the system call to resolve a certain value did not succeed. See accompanied error message for more detailed information.
- 4** usage displayed.
- 99** functionality not implemented yet.

FILES

The files involved to resolve the information are operating system dependent.

/var/adm/utmp
uptime on HP-UX.

EXAMPLES

The following examples is the result if all commands are executed at the same time.

0) system information

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: uname -a

Linux rh7mzv7t001 2.4.7-10 #1 Thu Sep 9 00:25:31 EDT 2006 i686 unknown
```

1) plain output

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: sysinfo -a

Date: 2006-09-30
Time: 00:25:31
Seconds since Epoch: 1159568731
Uptime: 4days, 1hours, 26minutes, 22seconds
Load Avgs: 1min(22240) 5min(6816) 15min(1952)
Total Ram: 513304k      Free: 76808k
Shared Ram: 0k
Buffered Ram: 104016k
Total Swap: 409616k    Free: 409616k
Number of processes: 78
```


2) uptime

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: sysinfo -u

2006-09-30 00:25:31;1159568731;350782

[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: uptime

12:25am up 4 days,  1:26,  2 users,  load average: 0.34, 0.10, 0.03
```

3) system load averages

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: sysinfo -l

2006-09-30 00:25:31;1159568731;22240;6816;1952
```

4) RAM

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: sysinfo -r

2006-09-30 00:25:31;1159568731;525623296;78651392;0;106512384
```

5) swap

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: sysinfo -s

2006-09-30 00:25:31;1159568731;419446784;419446784
```

6) processes

```
[ /opt/edrc/bin ]
[ root@rh7mzv7t001 ][*edrc*/bash]: sysinfo -n

2006-09-30 00:25:31;1159568731;78
```

SEE ALSO

ctime(3), date(1), edrcintro(1), sysinfo(2), sed(1), time(2), timer(1), uptime(1), utmp(5)

NOTES

The **sysinfo** command was derived from the source code example **sysinfo.c** provided by <detour@metalshell.com> on <http://www.metalshell.com> and has been enhanced by Christian Walther. See <http://www.metalshell.com> for additional coding examples.

BUGS

The field delimiter is fixated to `;` and is therefore not changeable with a command line option, yet. As a workaround the following command sequence can be issued:

```
sysinfo -u | sed -e 's/;/ delimiter /g'
```

Not all options or functions are implemented on all operating systems. The table below lists the options currently implemented for a certain operating system:

| OSID | -u | -l | -r | -s | -n |
|---------|-----|-----|-----|-----|-----|
| Linux | YES | YES | YES | YES | YES |
| HP-10 | YES | NO | NO | NO | NO |
| HP-11 | YES | NO | NO | NO | NO |
| HP-11i | YES | NO | NO | NO | NO |
| Solaris | YES | NO | NO | NO | NO |

AUTHOR

sysinfo was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

syscp – copy system file to sysconfig repository

SYNOPSIS

edrc/bin/syscp [**-h**]

syscp *file* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

Copy an (active) file on the system to the **sysconfig** repository that has been used to configure the system.

If a file is in multiple repositories it can be interactively chosen to which repository the file should be copied.

If a system file does not differ (content or permissions) from the file residing in the repository a **WARNING** is displayed.

Each file copy has to be acknowledged.

If the system file differs from the repository file and the repository contains "replacement tags" in the format **@[-_A-Z]+@** (e.g. **@HOSTNAME@** in a **hosts** file) as used in **post_exec** and **pre_exec** scripts, a warning is displayed. Most likely it is not wished to overwrite such a repository file.

If a new file is added to a repository and a part of the path on the system is a symlink that is not already added to the repository, the new path to the file and the file is not added to the repository. This is a safety precaution. If you encounter this situation you most likely want to add the original path to the repository instead of the path thru the symlink.

An example of such a situation is to add **/etc/init.d/nginx** to the repository, where the **init.d** directory is on some systems a symlink to **rc.d/init.d**. In this case you should add **/etc/rc.d/init.d/nginx** to the repository.

OPTIONS

-h usage message.

file ... list of files to be copied to the repository.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 2** operating system is not supported, yet. See **osid**(3) if you get this error.
- 4** usage printed.
- 5** command aborted pressing <Ctrl>+<C> or using **kill**(1).

FILES

<root_dir>/EDRC.sysconfig <root_dir>/EDRC.sysconfig.<name>
 settings file which contains information about current active configuration. This file is read when using **sysconfig-p**, **sysvi**, **syscp** and **sysdiff**.

EXAMPLES**1) usage example****1.1) display the applied configurations**

```
[ / ]
[ root@acme002 ][*edrc*/ksh]: sysconfig -p

sysconfig - configure a Unix/Linux system, by Chr. Walther

/ -

CONFIG:      production.acme.ch
NAME:
OSID:        HP-11
SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch
ROOTDIR:     /
BACKUPDIR:   /opt/edrc/var/backup
BACKUPFILE:  2005-07-12_20.30.04__sysconfig.cpio.gz
DESCRIPTION: PRODUCTION SERVER CONFIGURATION for Operating System
EXECUTED:    Tue Jul 12 20:30:04 METDST 2005

/ - base

CONFIG:      acme.ch
NAME:        core
OSID:        HP-11
SOURCE:      /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch
ROOTDIR:     /
BACKUPDIR:   /opt/edrc/var/backup
BACKUPFILE:  2004-11-25_14.54.06__sysconfig.cpio.gz
DESCRIPTION: BASE SERVER CONFIGURATION for Operating System
EXECUTED:    Thu Nov 25 14:54:06 MET 2004
```

1.2) use of syscp

```
[ /etc ]
[ root@acme002 ][*edrc*/ksh]: syscp auto_apl services auto_dat ifconfig.muxids /etc/hosts \
                             /etc/protocols /etc/init.d/nginx /var/opt/perf/parm

copy system file to '/opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/auto_apl'? <yn> [n] :y
syscp-INFO: system file copied to '/opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/auto_apl'
syscp-HINT: do not forget to distribute the repository

syscp-WARNING: no difference to file '/opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/services'
copy system file to '/opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/services'? <yn> [n] :

syscp-ERROR: file 'auto_dat' does not exist
```

```

syscp-WARNING: file 'ifconfig.muxids' is currently not member of a repository
syscp-INFO: multiple possible repositories for file 'ifconfig.muxids':
1) /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/ifconfig.muxids
2) /opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/ifconfig.muxids
select num:2
add system file as '/opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/ifconfig.muxids'? <yn> [n] :y
syscp-INFO: system file added as '/opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/ifconfig.muxids'
syscp-HINT: do not forget to distribute the repository

syscp-INFO: multiple matches for file '/etc/hosts':
1) /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/hosts
2) /opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/hosts
select num:1
syscp-WARNING: replacement tags in repository file '/opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/hosts'
copy system file to '/opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/hosts' ? <yn> [n] :

syscp-INFO: multiple matches for file '/etc/protocols':
1) /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/protocols
2) /opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/protocols
select num:1
copy system file to '/opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/protocols' ? <yn> [n] :y
syscp-INFO: system file copied to '/opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/protocols'
syscp-HINT: do not forget to distribute the repository

syscp-WARNING: file '/etc/init.d/nginx' is currently not member of a repository
syscp-INFO: multiple possible repositories for file '/etc/init.d/nginx':
1) /opt/edrc/scripts/ACME-sat/sysconfig/acme.ch/HP-11/root/etc/init.d/nginx
2) /opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/etc/init.d/nginx
select num:2
syscp-ERROR: system file path '/etc/init.d -> rc.d/init.d' is a symlink, handle it first

copy system file to '/opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/var/opt/perf/parm'? <yn> [n] :y
syscp-INFO: system file copied to '/opt/edrc/scripts/ACME-sat/sysconfig/production.acme.ch/HP-11/root/var/opt/perf/parm'
syscp-HINT: do not forget to distribute the repository

```

SEE ALSO

edrcintro(1), filelink(1), sysconfig(1), sysvi(1), sysdiff(1)

NOTES

-

BUGS

-

AUTHOR

syscp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sysdiff – compare system file(s) with file(s) in sysconfig repository

SYNOPSIS

edrc/bin/sysdiff [**-h**]

sysdiff [**-v**] *file* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

list if an (active) file on the system applied with **sysconfig** differs to the file(s) in the related repository. The files are considered as different if the content or the filepermissions differ. To obtain a plain list of files that differ from the applied file, redirect the output to **stderr**.

OPTIONS

-h usage message.

-v verbose output.

file ... system file(s) to compare to the related file in a **sysconfig** repository.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 operating system not supported.

4 usage printed.

5 the **sysdiff** command was aborted.

FILES

<root_dir>/EDRC.sysconfig<.name>

settings file(s) that contains information about current active configuration(s). This file is read when using **sysconfig -p**, **syscp**, **sysvi** and **sysdiff**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **filelink(1)**, **sysconfig(1)**, **syscp(1)**, **sysvi(1)**

NOTES

-

BUGS

-

AUTHOR

sysdiff was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

syspoll – poll systems to execute a query

SYNOPSIS

edrc/bin/syspoll [**-h**]

syspoll [**-c** *configfile*] [**-r** *retries[:interval]*] **-a** { **start** | **retry** } **-j** *jobname* [**-s** *setname*] [**-t** *hostlist*] [**-o** *outputfile*]

syspoll [**-c** *configfile*] **-a** { **list_retry** } **-j** *jobname* [**-s** *setname*]

syspoll [**-c** *configfile*] **-a** { **list_jobs** | **list_sets** | **list_protos** | **list_perf** }

syspoll [**-c** *configfile*] **-a** { **print_log** [**-b** *begin_date*][**-e** *end_date*] }

AVAILABILITY

WA2L/edrc

DESCRIPTION

poll systems and execute a query.

The main purpose is to centrally control report data collection for different target operating systems.

syspoll writes all command status and information output to **stderr** and only the job header and the query results to **stdout** respectively the *outputfile* if specified.

The following phases are traversed when a **syspoll** *job* is executed (**start** or **retry**) in a certain *set*:

1) **general information**

print general information about the environment where **syspoll** is started, the configuration and the impact of some configuration- and command line options.

2) **set information**

print information about the set.

3) **job information**

print information about the job.

4) **job header**

write the job header as specified in the **JOB_HEADER** setting in **VARDIR/jobs/<job-name>/job.cfg** to the *outputfile* (if specified) and **stdout** if the action **-a start** is selected.

If if the action **-a retry** is selected, the job header is not written (again) to the *outputfile* and is printed for reference purposes to **stderr** only.

5) connect to systems

connect to all systems as specified in the system target list using **-t hostlist**, the retry file or the **POLL_HOSTLIST** setting in the **etc/syspoll.cfg** file depending on the command line options selected.

5.1) resolve connection information

try to resolve connection information, as the IP address, of the system to connect to. If the IP address cannot be resolved, no connection is possible (of course).

5.2) probe connection

If the IP address could be resolved in the previous step, probe the connection to the target system using **is_up(3)** internally.

5.3) probe operating system

when the connection has been probed successfully, probe the operating system of the target system. This is actually the first test to check if a login to the target system is possible respectively, depending on the protocol, information can be queried from the target system.

5.4) handle query

if the operating system has been probed successfully in the previous step, handle the job query.

5.4.1) deploy query payload

if there are payload files present for the job, deploy (copy) the payload files to the target system. Not all protocols might be capable to deploy payload files (see: **lib/syspoll/protocols/<VARIANT>/<PLUGIN>**).

Payload files are additional programs, scripts or other data that is needed by the query to do its work and are not present on the target system.

5.4.2) execute query

if a query is defined, execute the query on/for the target system. When queries are written it is important to output all relevant query result data to **stdout**. Data/information printed to **stderr** will be seen on the screen but not saved in the *outputfile*.

If a query could be executed **syspoll** rates the related target system as processed successfully and it is removed from the retry list.

5.4.3) clear query payload

remove the deployed files from the target systems again.

6) execution summary

print an execution summary of the *job*.

As soon as it is possible, **syspoll** forecasts a completion of the entire job run based on the elapsed time, the total number of systems to process and the number of already processed target systems. The forecast is adjusted constantly.

syspoll has a plugin mechanism, where new protocols or new variants of protocols can be integrated.

New protocols might be SNMP or a database plugins to connect to PostgreSQL, MySQL, MSSQL or Oracle.

Protocol variants are existing protocols, but customized in the fine details of the connection method and the execution of queries. This might be the case if in a certain compliance situation it is only allowed to execute queries on remote systems using **sudo**(8). For this a variant of an existing protocol can be created without adjusting jobs.

OPTIONS

-h usage message. Here the revision of **syspoll** is also displayed.

-c *config_file*
configuration file.

-r *retries[:interval]*
number of retries and interval between two immediate connection retries in seconds.

-s *setname*
set name.

A *set* is a handle to specify some specific additional general credentials that are valid for a job; as the login credentials used to execute the query and the protocol variant used. With sets it is possible to write a job **query** and use it without modifications for different target customer environments even if the user names, passwords and the details to connect to the systems change.

When a job is started first with the *setname* **ACME** and then with **Riptide**, for each *set* an own retry list is maintained. It is intended to use the customer name as *setname* or a combination of the customer name and the targeted customer environment, as: **ACME.TEST**.

The default *setname* is the customer name of the environment where **syspoll** is started as returned by **server_environment -C** or **default** if the set configuration for the customer name does not exist.

A *set* is represented by a configuration file in the **var/syspoll/sets/** directory.

See also: **server_environment(3)**.

-a action:

start -j *jobname* [**-s** *setname*] [**-t** *hostlist*]
start a job for a given *set*.

The job header as specified in the **JOB_HEADER** setting in **VARDIR/jobs/<job-name>/job.cfg** is written once to the *outputfile* at the beginning of the job execution when the **start** action is executed.

retry -j *jobname* [**-s** *setname*] [**-t** *hostlist*]
retry job on failed systems for a given *set*.

The job header as specified in the **JOB_HEADER** setting in **VARDIR/jobs/<job-name>/job.cfg** is not written to the *outputfile* when the **retry** action is executed.

list_retry -j *jobname* [**-s** *setname*]
print current retry target list for a *job* for a given *set*.

list_jobs
list all defined jobs.

list_sets
list all defined sets.

list_protos
list all available protocols and protocol variants.

Currently the protocols SSH (secure shell) and WMI (Windows Management Instrumentation) are distributed with WA2L/edrc for **syspoll**.

list_perf
list performance statistics of past job runs.

print_log [-b *begin_date*] [**-e** *end_date*]
print the **syspoll** log file between the *begin_date* and the *end_date*.
If the *begin_date* is not specified, the log file is printed from the beginning to the *end_date*. If the *end_date* is not entered, the log file is printed from the *begin_date* to the end. If neither of the two dates is specified the whole log is printed.

-j *jobname*
name of a job to be executed.

-t *hostlist* list of target hosts. This list can be a comma separated list (as returned by **hostlist -l**), a file containing a list of hosts or a hostgroup name (*@HOSTRGOUP*).

The file has to be specified with an absolute path name.

When a dash is specified as *hostlist* (example: **syspoll -a start -j poll -t -**) the system target list is read from **stdin**.

-o *outputfile*

output file where the query results are saved. It is also possible to capture the query results by redirecting the **stdout** output of **syspoll**.

The **outputfile** has to be specified as an absolute path name. See also: **rel2abs(3)** to see how to convert a relative path name to an absolute.

-b *begin_date*

begin date in the military format **YYYY-MM-DD**. To compute dates in this format, see: **input(3)**, **seconds(3)**, **timer(1)**, **today(3)**, **tomorrow(3)**, **yesterday(3)**.

-e *end_date*

end date in the military format **YYYY-MM-DD**. To compute dates in this format, see: **input(3)**, **seconds(3)**, **timer(1)**, **today(3)**, **tomorrow(3)**, **yesterday(3)**.

ENVIRONMENT

\$SYSPOLL_CONFIGFILE

configuration file of **syspoll**. The **-c** *configfile* command line option has preference.

\$PRINT_FIT2WIDTH

If this environment variable is not set to *False*, the output of lists is limited to the current width of the terminal window and rows extending the window width are marked with '>>'.

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 1 | configuration file does not exist. |
| 2 | <i>setname</i> does not exist. |
| 3 | <i>jobname</i> does not exist. |
| 4 | Usage printed. |
| 5 | syspoll aborted pressing <Ctrl>+<C>. |
| 6 | cannot write to <i>outputfile</i> as specified in -o <i>outputfile</i> . |
| 7 | protocol variant as specified in SET_PROTOCOL_VARIANT setting in the set configuration file <i>setname.cfg</i> does not exist |

- 8** cannot write to logfile.
- 9** cannot write to **VARDIR**.
- 11** temporary directory could not be claimed or created in **/var/tmp**. Check the system temporary directory **/var/tmp** if you get this error, it is an indicator of system intrusion.

FILES

The **VARDIR** can be defined in the **syspoll.cfg** file. Default is **edrc/var/syspoll/**.

etc/syspoll.cfg

default **syspoll** configuration file.

lib/syspoll/protocols/<VARIANT>/<PLUGIN>

protocol plugins.

lib/syspoll/protocols/edrc/<PLUGIN>

default protocol plugins that profit from the connection setup of WA2L/edrc, where the key files are located in the **var/connection/security/** directory.

var/cache/syspoll/

cache files, do not edit them by hand.

var/log/syspoll.log

logfile of **syspoll**.

VARDIR/jobs/<jobname>/

directory containing the definition of a job.

VARDIR/jobs/<jobname>/job.cfg

some global settings for the job, as **JOB_DESCRIPTION** and **JOB_HEADER**.

VARDIR/jobs/<jobname>/<OSDIR>/<PROTOCOL>/query

<OSDIR> ::= **<os_name>.<os_release>.<os_machine>** | **<os_name>.<os_release>** | **<os_name>** | **default**

Query to be executed on a given operating system and the related protocol (**<PROTOCOL>**).

The following variables are exported to the query execution: **JOB_LOCALUSER**, **JOB_REMOTEUSER**, **JOB_PAYLOAD**, **JOB_QUERY**, **JOB_PROTOCOL**, **JOB_PASSWORD**, **JOB_LOCALTEMP**, **JOB_POSTQUERY**, **JOB_HOSTNAME**, **AWK**, **ECHO**, **TAIL** and **OSID**.

VARDIR/jobs/<jobname>/<OSDIR>/<PROTOCOL>/payload/

directory containing additional files that are temporarily needed on the target system to execute the query.

VARDIR/locks/

lockfiles, do not edit them by hand.

VARDIR/sets/<setname>.cfg

set configuration file.

VARDIR/sets/{<SERVER_ENVIRONMENT_CUSTOMER>.cfg[default.cfg]}

default set configuration file. If **-s setname** is not specified the *setname* defaults first to the customer name of the server environment (<SERVER_ENVIRONMENT_CUSTOMER>) as returned by **server_environment -C** where **syspoll** is started, if that **.cfg** does not exist to **default**.

VARDIR/state/

state information of **syspoll**, do not edit those files by hand.

VARDIR/state/<setname>.<jobname>.retry

retry file of systems that were not reached for a *jobname* in a given set (*setname*). The retry file contains all target systems that were not reachable plus all remaining systems if **syspoll** is aborted.

The retry file is reset when **-a start** is invoked.

When **-a retry** is issued once or multiple times, systems with a successful retry are removed from the retry file, unsuccessful systems remain in the file.

VARDIR/state/performance

record of job execution durations and success rates.

EXAMPLES

As always in WA2L/edrc, if the setup in **server_environment.cfg** and **hostlist.cfg** is done, the command runs within the environment without specifying targets. Therefore the options **-s setname** and **-t hostlist** can be omitted when the systems that are polled reside in the same environment as the host where **syspoll** is executed.

Thus, the most simplest invocation of **syspoll** is:

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start -j bdf

[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a retry -j bdf
```

The two example jobs referenced in this section are *probe* and *bdf*, where *probe* tests connections to target systems without executing a real query and *bdf* resolves the file system disk space usage as done by **df(1)** or **bdf(1)** and returns the result as csv file having the format **'HOST;FILESYS-TEM;TOTAL;FREE;USED;%USED;MOUNTPOINT;'**.

1) start the 'probe' job to check target connectivity

The *probe* job is a job that is considered to be part of the **syspoll** command (therefore do not change it, create a copy if you need a modified version of it). This job does not have a real

query, the purpose of this job is simply to test connections to the given targets and verify that the connection settings are correct.

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j probe \
-t acme002,acme003,acme004
```

2) start 'bdf' job and save query results to new output file

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j bdf \
-t acme002,acme003,acme004 \
-o /tmp/bdf.csv
```

3) retry 'bdf' job for failed targets and add query results to existing output file

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a retry \
-s ACME -j bdf \
-o /tmp/bdf.csv
```

4) retry 'bdf' job for additional targets and add query results to existing output file

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a retry \
-s ACME -j bdf \
-t acme005,acme006 \
-o /tmp/bdf.csv
```

5) list current retry list for job 'bdf' in 'ACME' set

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a list_retry \
-s ACME -j bdf
```

6) start a job on targets defined in hostgroup '@WEB'

See also: Example 10).

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a list_retry \
-s ACME -j bdf \
-t @WEB
```

7) start a job on targets defined in a file

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j probe \
-t /tmp/hostlist.txt
```

8) start a job on targets received from stdin

There is a file having pairs of hostnames and customer names separated by whitespaces:

```
acme001    ACME
acme002    ACME
websrv01   Bedrock
erp5       Bedrock
acme008    ACME
:
```

Run the *job* only on targets that are used for the customer 'ACME':

```
[ / ]
[ root@acme001 ][*eshell*/bash] \
awk '/\sACME$/ {print $1}' < /tmp/assets.lst | \
syspoll -a start -s ACME -j probe -t -
```

9) start a job on targets defined in a sqlite database

Run the *job* only on targets that are used for the customer 'ACME':

```
[ / ]
[ root@acme001 ][*eshell*/bash] sqlite /tmp/assets.db \
"select name from hosts where customer='ACME'" | \
syspoll -a start -s ACME -j probe -t -
```

10) start a job on targets defined in hostlist

See also: Example 6).

Using the setting `USE_HOSTLIST_DAT=True` and `CMAN_ENVIRONMENT=Highlander.ADMIN` in `hostlist.cfg` and when `syspoll` is started on a host that is member of the central management environment *Highlander.ADMIN* **hostlist(3)** can be used to address all targets of a certain customer.

Check if logged on to central management environment *Highlander.ADMIN* :

```
[ / ]
[ root@adm001 ][*eshell*/bash]: whereami

LOGGED ON TO SERVER ENVIRONMENT:

Customer           : Highlander
Environment         : ADMIN
Description         : Beringen, EDRC Development
```

Run the *job* only on targets that are used for the customer 'ACME':

```
[ / ]
[ root@adm001 ][*eshell*/bash] syspoll -a start \
-s ACME -j bdf -t 'hostlist -l -g @ACME.ALL'
```

11) start 'bdf' job and directly compute query results

Print a formatted list from csv output:


```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j bdf | print_list
```

Print a formatted list from csv output with a reduced set of columns:

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j bdf \
-t acme002,acme003,acme004 | \
select_columns ";" "HOST;MOUNTPOINT;TOTAL" | print_list
```

Omit status information output during data collection:

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j bdf \
-t acme002,acme003,acme004 \
2>/dev/null | \
select_columns ";" "HOST;MOUNTPOINT;TOTAL" | print_list
```

12) start a job and provide collected query results on WA2L/edrc:report

See also: Example 13), NOTES.

```
[ / ]
[ root@acme001 ][*eshell*/bash] syspoll -a start \
-s ACME -j bdf \
-o ~edrc/var/www/report/006/ACME/bdf/acme__bdf_'date +%Y%m%d'.csv
```

13) crontab entry to handle monthly data collection and provision of results to WA2L/edrc:report

See also: Example 12), NOTES.

In this example the collected query results are saved to the **\$HOME** directory of the user **'report'** what allows to access the report output directories thru **~report** from shell, scripts etc. independent of the actual userhome location.

Preparations to provide the collected data with WA2L/edrc:report:

1) create the directory structure to save the collected data

```
[ / ]
[ root@acme001 ][*eshell*/bash] mkdir ~report/ACME/bdf/

[ / ]
[ root@acme001 ][*eshell*/bash] chmod 755 ~report/ACME/bdf/
```

2) create a symbolic link from WA2L/edrc:report to the report data

```
[ / ]
[ root@acme001 ][*eshell*/bash] symlink ~report ~edrc/var/www/report/006
```

Create the crontab entry:

```
1 0 25 * * [ -d ~report/ ] && ~edrc/bin/syspoll -a start -s ACME -j bdf \
-o ~report/ACME/bdf/acme__bdf_'date +%Y%m%d'.csv > /dev/null 2>&1

1 2-20 25 * * [ -d ~report/ ] && ~edrc/bin/syspoll -a retry -s ACME -j bdf \
-o ~report/ACME/bdf/acme__bdf_'date +%Y%m%d'.csv > /dev/null 2>&1
```

In this definition the job is started once at **00:01** on every 25th of every month.

Then on the same day the job retries to query the unsuccessful targets every hour between 02:01 and 20:01.

Due to the fact, that **syspoll** locks a *job* for a *set* during execution it also does not create a mess if this job here runs longer then an hour.

The output is saved to the **~report/ACME/bdf/** directory.

The output file uses the reverse date as filename element using the **date(1)** command **acme__bdf_`date +%Y%m%d`.csv** what results in a filename of **acme__bdf_20130225.csv** when executed on February 25th 2013 (see also: <http://www.tugsat.tugraz.at/launch>).

14) example output of syspoll

Start the *bdf* job for set *ACME* and save job query results to **/tmp/diskspace.csv**:

```
[ / ]
[ root@acme001 ][*eshell*/bash]: syspoll -a start -j bdf \
-s ACME -t acme001,acme002,acme003,acme004,acme005,acme006 -o /tmp/diskspace.csv

syspoll - poll systems, by Chr. Walther

poll systems ...
  general information ...
    Customer ..... : WA2L
    Environment ..... : [ EDRC_DVLP ] Beringen, EDRC Development
    Action ..... : start
    Hostname ..... : acme001
    Who ..... : root
    Operating System . : Linux acme001 3.13.0-45-generic #74-Ubuntu SMP
    Date ..... : Sat Feb 21 13:08:06 CET 2015
    Revision ..... : 13
    Configfile ..... : /opt/edrc/etc/syspoll.cfg
    VARDIR ..... : /opt/edrc/var/syspoll
    RETRY_COUNTER .... : 2
    RETRY_INTERVAL ... : 2
    OSID ..... : Linux
  done.
  set information ...
    setname ..... : ACME
    description ..... : ACME Hardware Corporation set.
    revision ..... : 00
    protocol variant . : edrc
  done.
  job information ...
    jobname ..... : bdf
    description ..... : disk space
    revision ..... : 00
    outputfile ..... : /tmp/diskspace.csv (create)
  done.
  job header ...
    HOST;FILESYSTEM;TOTAL;FREE;USED;%USED;MOUNTPOINT;
  done.
  connect to systems ...
    host 'acme001' [8/8] ...
      connection information ...
        hostname ..... : acme001
        hostaliases ..... :
        ipaddress ..... : 10.10.10.11
      done.
      probe connection ...
        up, success.
      done.
      probe operating system ...
        try protocol ... ( SSH )... done.
        os name ..... : Linux
        os release ..... : 3.13.0-45-generic
        os machine ..... : x86_64
        protocol ..... : SSH
        local user ..... : edrc
        remote user ..... : root
```

```

done.
handle query ...
  deploy query payload ...
    no payload to deploy, skip.
  done.
  execute query ...
    acme001;/dev/sda6;41151808;17058308;21980068;53.412156;/;
    acme001;none;4;4;0;0.000000;/sys/fs/cgroup;
    acme001;udev;1013352;1013348;4;0.000395;/dev;
    acme001;tmpfs;204832;203556;1276;0.622950;/run;
    acme001;none;5120;5120;0;0.000000;/run/lock;
    acme001;none;1024144;1023920;224;0.021872;/run/shm;
    acme001;none;102400;102364;36;0.035156;/run/user;
    acme001;/dev/sda1;999320;720620;209888;21.003082;/boot;
    acme001;/dev/sda2;51475068;2071508;46765736;90.851237;/home;
  done.
  clear query payload ...
    payload cleared.
  done.
done.
host 'acme002' [7/8] ...
  estimated job completion: 2015-02-21 13:08:24
  connection information ...
    hostname ..... : acme002
    hostaliases ..... :
    ipaddress ..... : 10.10.10.12
  done.
  probe connection ...
    up, success.
  done.
  probe operating system ...
    try protocol ... ( SSH )... done.
    os name ..... : SunOS
    os release ..... : 5.10
    os machine ..... : i86pc
    protocol ..... : SSH
    local user ..... : edrc
    remote user ..... : root
  done.
  handle query ...
    deploy query payload ...
      no payload to deploy, skip.
    done.
    execute query ...
      <<< OUTPUT TO BE ADDED >>>
    done.
    clear query payload ...
      payload cleared.
    done.
  done.
done.
host 'acme003' [6/8] ...
  estimated job completion: 2015-02-21 13:08:24
  connection information ...
    hostname ..... : acme003
    hostaliases ..... :
    ipaddress ..... : 10.10.10.13
  done.
  probe connection ...
    up, success.
  done.
  probe operating system ...
    try protocol ... ( SSH )... done.
    os name ..... : AIX
    os release ..... : 6.1
    os machine ..... : powerpc
    protocol ..... : SSH
    local user ..... : edrc
    remote user ..... : root
  done.
  handle query ...
    deploy query payload ...
      no payload to deploy, skip.
    done.
    execute query ...
      <<< OUTPUT TO BE ADDED >>>
    done.
    clear query payload ...
      payload cleared.
    done.
  done.

```

```

done.
done.
host 'acme004' [5/8] ...
  estimated job completion: 2015-02-21 13:08:24
  connection information ...
    hostname ..... : acme004
    hostaliases ..... :
    ipaddress ..... : 10.10.10.14
done.
probe connection ...
  up, success.
done.
probe operating system ...
  try protocol ...( SSH )... done.
  os name ..... : HP-UX
  os release ..... : B.11.31
  os machine ..... : ia64
  protocol ..... : SSH
  local user ..... : edrc
  remote user ..... : root
done.
handle query ...
  deploy query payload ...
    no payload to deploy, skip.
  done.
  execute query ...
    <<< OUTPUT TO BE ADDED >>>
  done.
  clear query payload ...
    payload cleared.
  done.
done.
done.
host 'acme005' [4/8] ...
  estimated job completion: 2015-02-21 13:08:22
  connection information ...
    hostname ..... : acme005
    hostaliases ..... :
    ipaddress ..... : 10.10.10.15
done.
probe connection ...
  up, success.
done.
probe operating system ...
  try protocol ...( SSH WMI )... done.
  os name ..... : Windows
  os release ..... : 2012
  os machine ..... :
  protocol ..... : WMI
  local user ..... : edrc
  remote user ..... : root
done.
handle query ...
  deploy query payload ...
    no payload to deploy, skip.
  done.
  execute query ...
    <<< OUTPUT TO BE ADDED >>>
  done.
  clear query payload ...
    payload cleared.
  done.
done.
done.
host 'acme006' [3/8] ...
  estimated job completion: 2015-02-21 13:08:22
  connection information ...
    hostname ..... : acme006
    hostaliases ..... :
    ipaddress ..... :
done.
probe connection ...
  cannot resolve ip address, skip.
fail.
probe operating system ...
  connection probe failed, skip.
fail.
handle query ...
  operating system probe failed, skip.
fail.
fail.

```

```
host 'acme007' [2/8] ...
  estimated job completion: 2015-02-21 13:08:23
  connection information ...
    hostname ..... : acme007
    hostaliases ..... :
    ipaddress ..... : 10.10.10.17
  done.
  probe connection ...
    cannot connect, wait . retry
    cannot connect, wait . retry
    not up, fail.
  fail.
  probe operating system ...
    connection probe failed, skip.
  fail.
  handle query ...
    operating system probe failed, skip.
  fail.
fail.
host 'acme008' [1/8] ...
  estimated job completion: 2015-02-21 13:08:54
  connection information ...
    hostname ..... : acme008
    hostaliases ..... : www.acme.com www.acme.ch
    ipaddress ..... : 10.10.10.18
  done.
  probe connection ...
    up, success.
  done.
  probe operating system ...
    try protocol ...( SSH WMI )... done.
    os name ..... : unknown
    os release ..... : unknown
    os machine ..... : unknown
    protocol ..... : unknown
  fail.
  handle query ...
    operating system probe failed, skip.
  fail.
fail.
done.
execution summary ...
  failure for: acme006,acme007,acme008
  unknown for:
  summary: success/failure/unknown/total 5/3/0/8
done.
done.
```

List targets in retry list:

```
[ / ]
[ root@acme001 ][*eshell*/bash]: syspoll -a list_retry -j bdf -s ACME

syspoll - poll systems, by Chr. Walther

HOSTNAME
-----
acme006
acme007
acme008
(1)
```

List performance information:

```
[ / ]
[ root@acme001 ][*eshell*/bash]: syspoll -a list_perf

syspoll - poll systems, by Chr. Walther
```

| SET | JOB | t/RUN | t/TARGET | SUCCESS_RATE | RUNS | TARGETS | LAST_RUN |
|---------|-------|---------|----------|--------------|------|---------|---------------------|
| ACME | bdf | 0:03:35 | 0:00:26 | 62.50 % | 1 | 8 | 2015-02-21 13:08:06 |
| WA2L | bdf | 0:00:08 | 0:00:02 | 85.71 % | 2 | 7 | 2015-02-21 13:06:39 |
| WA2L | probe | 0:00:26 | 0:00:07 | 15.00 % | 6 | 20 | 2015-02-21 11:43:12 |
| default | probe | 0:00:01 | 0:00:00 | 0.00 % | 1 | 3 | 2015-02-20 23:54:50 |

(4)

Retry job on failed targets and add additional query results to /tmp/diskspace.csv :

```
[ / ]
[ root@acme001 ][*eshell*/bash]: syspoll -a retry -j bdf -s ACME -o /tmp/diskspace.csv

syspoll - poll systems, by Chr. Walther

poll systems ...
  general information ...
    Customer ..... : WA2L
    Environment ..... : [ EDRC_DVLP ] Beringen, EDRC Development
    Action ..... : retry
    Hostname ..... : acme001
    Who ..... : root
    Operating System . : Linux acme001 3.13.0-45-generic #74-Ubuntu SMP
    Date ..... : Sat Feb 21 13:51:09 CET 2015
    Revision ..... : 13
    Configfile ..... : /opt/edrc/etc/syspoll.cfg
    VARDIR ..... : /opt/edrc/var/syspoll
    RETRY_COUNTER .... : 2
    RETRY_INTERVAL ... : 2
    OSID ..... : Linux
  done.
  set information ...
    setname ..... : ACME
    description ..... : ACME Hardware Corporation set.
    revision ..... : 00
    protocol variant . : edrc
  done.
  job information ...
    jobname ..... : bdf
    description ..... : disk space
    revision ..... : 00
    outputfile ..... : /tmp/diskspace.csv (append)
  done.
  job header ...
    HOST;FILESYSTEM;TOTAL;FREE;USED;%USED;MOUNTPOINT;
  done.
  connect to systems ...
    host 'acme006' [3/3] ...
      connection information ...
        hostname ..... : acme006
        hostaliases ..... :
        ipaddress ..... :
      done.
      probe connection ...
        cannot resolve ip address, skip.
      fail.
      probe operating system ...
        connection probe failed, skip.
      fail.
      handle query ...
        operating system probe failed, skip.
      fail.
    fail.
    host 'acme006' [2/3] ...
      estimated job completion: 2015-02-21 13:08:22
      connection information ...
        hostname ..... : acme007
        hostaliases ..... :
        ipaddress ..... : 10.10.10.16
      done.
      probe connection ...
        up, success.
      done.
      probe operating system ...
        try protocol ... ( SSH )... done.
        os name ..... : Linux
        os release ..... : 3.13.0-45-generic
        os machine ..... : x86_64
        protocol ..... : SSH
        local user ..... : edrc
        remote user ..... : root
      done.
      handle query ...
        deploy query payload ...
          no payload to deploy, skip.
        done.
        execute query ...
          acme007;/dev/sda6;41151808;17060952;21977424;53.405731;/;
          acme007;none;4;4;0;0.000000;/sys/fs/cgroup;
          acme007;udev;1013352;1013348;4;0.000395;/dev;
          acme007;tmpfs;204832;203560;1272;0.620997;/run;
```

```

acme007;none;5120;5120;0;0.000000;/run/lock;
acme007;none;1024144;1023920;224;0.021872;/run/shm;
acme007;none;102400;102364;36;0.035156;/run/user;
acme007;/dev/sda1;999320;720620;209888;21.003082;/boot;
acme007;/dev/sda2;51475068;2068184;46769060;90.857694;/home;
done.
clear query payload ...
payload cleared.
done.
done.
done.
host 'acme008' [1/3] ...
estimated job completion: 2015-02-21 13:08:54
connection information ...
hostname ..... : acme008
hostaliases ..... : www.acme.com www.acme.ch
ipaddress ..... : 10.10.10.18
done.
probe connection ...
up, success.
done.
probe operating system ...
try protocol ... ( SSH WMI ) ... done.
os name ..... : unknown
os release ..... : unknown
os machine ..... : unknown
protocol ..... : unknown
fail.
handle query ...
operating system probe failed, skip.
fail.
fail.
done.
done.
execution summary ...
failure for: acme006
unknown for:
summary: success/failure/unknown/total 1/2/0/3
done.
done.

```

List targets in retry list:

```

[ / ]
[ root@acme001 ][*eshell*/bash]: syspoll -a list_retry -j bdf -s ACME

syspoll - poll systems, by Chr. Walther

HOSTNAME
-----
acme006
acme008
(1)

```

SEE ALSO

edrcintro(1), **edrcinit(1m)**, **csv2worksheet(3)**, **hostlist(3)**, **input(3)**, **is_up(3)**, **osid(3)**, **print_list(3)**, **rcmd(1)**, **rel2abs(3)**, **scp(1)**, **seconds(3)**, **server_environment(3)**, **ssh(1)**, **ssh-exec(1)**, **syspoll.cfg(4)**, **timer(1)**, **today(3)**, **tomorrow(3)**, **WA2Ledrc.report(1)**, **whereami(1)**, **wmic(1)**, **yesterday(3)**

NOTES

To serve reports with the web application '**WA2L/edrc:report - The Operating System Report Portal**' included in WA2L/edrc, write the reports to a directory structure having the convention **<YOUR_REPORTBASE>/<SERVER_ENVIRONMENT_CUSTOMER>/<reportname>/<outputfile>** and the world readable flag set. Then create a symbolic link using **symlink(1)** from **var/www/report/rpt/<NUMBER>** to **<YOUR_REPORTBASE>**.

See also: Example 12) and 13) in **EXAMPLES**, **WA2Ledrc:report(1)**, **edrcinit(1m)**, and '**http://hostname:9900 -> Help**' for more information about **WA2L/edrc:report**.

BUGS

Abort handling is fully tested under Linux only. Therefore when running **syspoll** on other operating systems, refrain from aborting a running **syspoll** session if possible. However, the expected side effects are not severe.

AUTHOR

syspoll was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

syspoll.cfg – configuration file for syspoll

SYNOPSIS

edrc/etc/syspoll.cfg
\$SYSPOLL_CONFIGFILE

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **syspoll** command.

FILEFORMAT

Rows starting with a **#** are considered as comments.

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the **=** and the **VALUE** are no spaces.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a **VALUE**.

OPTIONS

VARDIR location of the **edrc/var/syspoll** directory containing all state information, job files and set configurations. When the **VARDIR** is changed from the default location it has to be ensured that the whole structure of the **VARDIR** is present at the altered location. Ensure that the directory permissions are identical to the ones as set at the default location.

See also description in **syspoll(1)** concerning this topic.

Example: **VARDIR=/dat/syspoll/var/syspoll**

Default: **VARDIR=edrc/var/syspoll**

LOG Log output dir of syspoll. If you specify a relative path name the path is relative to the WA2L/edrc installation directory.

Example: **LOG=/var/opt/syspoll/log**

Default: **LOG=edrc/log**

POLL_HOSTLIST

Space separated list of hosts. This hosts will be polled if not specified on command line.

Example: POLL_HOSTLIST="hostlist -g @ALL"

Default: POLL_HOSTLIST='hostlist'

RETRY_COUNTER

Number of immediate connection retries.

Example: RETRY_COUNTER=10

Default: RETRY_COUNTER=2

RETRY_INTERVAL

Interval (pause between retries) of connection retries in seconds.

Example: RETRY_INTERVAL=10

Default: RETRY_INTERVAL=5

LOCK_TIMEOUT

timeout in seconds. This is the maximum lifetime of a job lock. This setting has to be adjusted only, if the data processing duration exceeds the default.

If the jobs are scheduled, the **LOCK_TIMEOUT** should not exceed the minimum scheduling interval.

Example: LOCK_TIMEOUT=10800

Default: LOCK_TIMEOUT=43200

SEE ALSO

edrcintro(1), **syspoll(1)**

NOTES

-

BUGS

-

AUTHOR

syspoll.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sysvi – edit a file in sysconfig repository

SYNOPSIS

edrc/bin/sysvi [**-h**]

sysvi *file* ...

svi [**-h**]

svi *file* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

edit the file in a sysconfig repository which is related to the local (active) file on the system.

If a file is in multiple repositories it can be interactively chosen in which repository the file should be edited.

OPTIONS

-h usage message.

file ... list of files to be edited in the repository.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 2 | operating system is not supported, yet. See osid (3) if you get this error. |
| 4 | usage printed. |
| 5 | command aborted pressing <Ctrl>+<C> or using kill (1). |

- 11** temporary directory could not be claimed in **/tmp**. Check the system if you get this error, it is an indicator for system intrusion.

FILES

<root_dir>/EDRC.sysconfig<.name>

settings file(s) that contains information about current active configuration(s). This file is read when using **sysconfig -p**, **syscp**, **sysvi** and **sysdiff**.

etc/exrc settings for the **vi** / **vim** editor.

EXAMPLES

-

SEE ALSO

edrcintro(1), **filelink(1)**, **sysconfig(1)**, **syscp(1)**, **sysdiff(1)**

NOTES

-

BUGS

-

AUTHOR

sysvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sys2html – collect system configuration and create a HTML report

SYNOPSIS

edrc/bin/sys2html [**-h**]

sys2html **-o** *outdir* [*user@*]*hostname*...

sys2html [*options*] **-o** *outdir* [*user@*]*hostname*...

sys2html [*options*] **-r** [*user@*]*hostname*... > *data.raw*

sys2html **-o** *outdir* - < *data.raw*

sys2html [**-c** *config_file*] **-l**

sys2html [**-c** *config_file*] **-p** *OSID*

options ::= [**-m**] [**-n** *name_resolver*] [**-i** *info_resolver*] [**-c** *config_file*] [**-s** *info_file*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **sys2html** command collects the system configuration into a 'System Documentation' HTML file.

It includes the collection of **cron** and **at**, installed hardware and software, filesystems, dump- and swap-configuration, LVM, network settings, kernel, system enhancements and applications.

The HTML 'System Documentation' report is also suited for viewing using **lynx**(1) on the command line.

sys2html aims to collect the same information as the well known **cfg2html**(1m) does, but without the need to install it on the (remote) systems where the configuration is collected.

To connect to the target system SSH (with **ssh-exec**) is used.

The generation of the HTML 'System Documentation' report is performed and saved on the system where the **sys2html** is invoked.

OPTIONS

-h usage message.

- c** *config_file* specify an alternate configuration collection configuration **sys2html.dat**(4) file.
- i** *info_resolver* custom host name to system description resolver. The *info_resolver* is used to dynamically resolve the system description.

If it cannot return a system description, the system description is looked-up in the **sys2html.inf** file, if possible.

The *info_resolver* must be an executable that receives a *hostname* as argument and returns the system description to **stdout**, whereas the output can consist of multiple lines.
- l** list currently configured operating systems (*OSID*) to collect system information from.
- m** force **MULTISTEP** query mode, where for each configuration row with defined *commands* an SSH connection to the target system is made in sequence.
- n** *name_resolver* custom host name to IP address resolver. The *name_resolver* is used if the IP address cannot be resolved by the system (using **getent hosts hostname** internally). It must be an executable that receives a *hostname* as argument and returns the resolved IP address as first field, whereas the fields are white-space separated.
- o** *outdir* output directory where to save the **HTML** 'System Documentation' reports.
- p** *OSID* print collection definition (**TITLE**, **COMMAND**) for the given *OSID*.
- r** print raw data of collected configuration to **stdout**.
- s** *info_file* specify an alternate central system information description **sys2html.inf**(4) file.
- read from **stdin**.
- data.raw* raw data returned by **sys2html** not formatted to **HTML**.
- user* user to connect to on the remote system. If no *user* is specified, this defaults to **root**.
- hostname* space separated list of systems where to remotely collect the system configuration.

ENVIRONMENT

-

EXIT STATUS

- | | |
|----------|---|
| 0 | no error. |
| 1 | cannot write to the <i>outdir</i> directory specified in the -o option. |
| 2 | operating system not supported. See osid (3) if you get this error. |
| 3 | on command line specified sys2html.cfg or sys2html.inf could not be read. |
| 4 | usage displayed. |

FILES

- etc/sys2html.dat**
configuration collection configuration file for **sys2html**(1m).
- etc/sys2html.inf**
central system information description file for **sys2html**(1m).
- /etc/cfg2html/systeminfo**
local system description file as known from the **cfg2html**(1m) command.

EXAMPLES

-

SEE ALSO

edrcintro(1), **cfg2html**(1m), **getent**(1), **osid**(3), **ssh-exec**(1), **sys2html.dat**(4), **sys2html.inf**(4), **lynx**(1), <https://www.cfg2html.com>

NOTES

The **sys2html** command is a simplified "clone" of the **cfg2html**(1m) command and it is highly inspired by it.

Most obvious differences to the **cfg2html** are, that **sys2html**:

- collects the system configuration from remote systems,
- allows to define the elements to be collected on the remote system in a configuration file and
- is currently configured to collect system information from **Linux** (OSID: Linux-64) and **Dell/EMC Data Domain** (OSID: DELLDataDomain) systems.

BUGS

The configuration collection of **sys2html** is not that sophisticated as the one from **cfg2html** due to the configuration file approach.

AUTHOR

sys2html was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sys2html.dat – configuration query definition for sys2html

SYNOPSIS

edrc/etc/sys2html.dat

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration query definition for the **sys2html** command.

This file specifies all configuration query commands to be executed on the target system and the **HTML** report structure.

FILEFORMAT

The fileformat is a list of definitions that have the format

OSID;TITLE;COMMAND

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

OSID operating system id of the target system. See also **osid**(3) and **rosid**(3).

TITLE section title or subtitle in the **HTML** report.

If no *COMMAND* is specified, the entry is treated as a section title.

Title entries starting with an **@** are options and its *COMMAND* entry is not executed on the target system.

Options are:

@MODE Query mode.

Set **SINGLESTEP** in **COMMAND** (is the default) to submit the whole configuration query definition as one command sequence to the target system.

MULTISTEP causes to submit each definition row separately to the target system. This mode is slower as **SINGLESTEP** and is only needed if the target system does not provide the **echo** command.

This mode can be forced using the **-m** option and can be used to debug query

configurations and identify a long running or hanging *COMMAND* .

@FILTER

this option allows to define a filter in **COMMAND** that is applied to the whole query result data and can be used to filter out unwanted texts.

@REPORTTITLE

this option allows to change the report title different to the default **System Documentation**.

COMMAND

command to query the configuration. It is allowed to use the semicolon character (;) in this field.

EXAMPLES

-

SEE ALSO

edrcintro(1), **osid(1m)**, **rosid(1m)**, **sys2html(1m)**, **sys2html.inf(4)**

NOTES

-

BUGS

-

AUTHOR

sys2html.dat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

sys2html.inf – system information configuration for sys2html

SYNOPSIS

edrc/etc/sys2html.inf

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the system information configuration for the **sys2html** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

HOSTNAME;DESCRIPTION;

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

HOSTNAME

host name of the target system.

The *HOSTNAME* is a regular expression.

DESCRIPTION

Description added to **'System Information'** in the **'General'** section if the hostname matches to the *HOSTNAME* regular expression.

Line breaks in the output can be added using the **\n** escape sequence.

EXAMPLES

-

SEE ALSO

edrcintro(1), **regexintro**(4), **sys2html**(1m), **sys2html.dat**(4)

NOTES

-

BUGS

-

AUTHOR

sys2html.inf was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`tcpdump` – dump traffic on a network

SYNOPSIS

```
tcpdump [ -adefnNOpqRStuvxX ] [ -c count ]
          [ -C file_size ] [ -F file ]
          [ -i interface ] [ -m module ] [ -r file ]
          [ -s snaplen ] [ -T type ] [ -w file ]
          [ -E algo:secret ] [ expression ]
```

DESCRIPTION

Tcpdump prints out the headers of packets on a network interface that match the boolean *expression*. It can also be run with the **-w** flag, which causes it to save the packet data to a file for later analysis, and/or with the **-r** flag, which causes it to read from a saved packet file rather than to read packets from a network interface. In all cases, only packets that match *expression* will be processed by *tcpdump*.

Tcpdump will, if not run with the **-c** flag, continue capturing packets until it is interrupted by a SIGINT signal (generated, for example, by typing your interrupt character, typically control-C) or a SIGTERM signal (typically generated with the **kill**(1) command); if run with the **-c** flag, it will capture packets until it is interrupted by a SIGINT or SIGTERM signal or the specified number of packets have been processed.

When *tcpdump* finishes capturing packets, it will report counts of:

packets “received by filter” (the meaning of this depends on the OS on which you’re running *tcpdump*, and possibly on the way the OS was configured - if a filter was specified on the command line, on some OSes it counts packets regardless of whether they were matched by the filter expression, and on other OSes it counts only packets that were matched by the filter expression and were processed by *tcpdump*);

packets “dropped by kernel” (this is the number of packets that were dropped, due to a lack of buffer space, by the packet capture mechanism in the OS on which *tcpdump* is running, if the OS reports that information to applications; if not, it will be reported as 0).

On platforms that support the SIGINFO signal, such as most BSDs, it will report those counts when it receives a SIGINFO signal (generated, for example, by typing your “status” character, typically control-T) and will continue capturing packets.

Reading packets from a network interface may require that you have special privileges:

Under SunOS 3.x or 4.x with NIT or BPF:

You must have read access to */dev/nit* or */dev/bpf**.

Under Solaris with DLPI:

You must have read/write access to the network pseudo device, e.g. */dev/le*. On at least some versions of Solaris, however, this is not sufficient to allow *tcpdump* to capture in promiscuous mode; on those versions of Solaris, you must be root, or *tcpdump* must be installed setuid to root, in order to capture in promiscuous mode. Note that, on many (perhaps all) interfaces, if you don’t capture in promiscuous mode, you will not see any outgoing packets, so a capture not done in promiscuous mode may not be very useful.

Under HP-UX with DLPI:

You must be root or *tcpdump* must be installed setuid to root.

Under IRIX with snoop:

You must be root or *tcpdump* must be installed setuid to root.

Under Linux:

You must be root or *tcpdump* must be installed setuid to root.

Under Ultrix and Digital UNIX/Tru64 UNIX:

Any user may capture network traffic with *tcpdump*. However, no user (not even the super-user) can capture in promiscuous mode on an interface unless the super-user has enabled promiscuous-mode operation on that interface using *pfconfig*(8), and no user (not even the super-user) can

capture unicast traffic received by or sent by the machine on an interface unless the super-user has enabled copy-all-mode operation on that interface using *pfconfig*, so *useful* packet capture on an interface probably requires that either promiscuous-mode or copy-all-mode operation, or both modes of operation, be enabled on that interface.

Under BSD:

You must have read access to */dev/bpf**.

Reading a saved packet file doesn't require special privileges.

OPTIONS

- a** Attempt to convert network and broadcast addresses to names.
- c** Exit after receiving *count* packets.
- C** Before writing a raw packet to a savefile, check whether the file is currently larger than *file_size* and, if so, close the current savefile and open a new one. Savefiles after the first savefile will have the name specified with the **-w** flag, with a number after it, starting at 2 and continuing upward. The units of *file_size* are millions of bytes (1,000,000 bytes, not 1,048,576 bytes).
- d** Dump the compiled packet-matching code in a human readable form to standard output and stop.
- dd** Dump packet-matching code as a C program fragment.
- ddd** Dump packet-matching code as decimal numbers (preceded with a count).
- e** Print the link-level header on each dump line.
- E** Use *algo:secret* for decrypting IPsec ESP packets. Algorithms may be **des-cbc**, **3des-cbc**, **blow-fish-cbc**, **rc3-cbc**, **cast128-cbc**, or **none**. The default is **des-cbc**. The ability to decrypt packets is only present if *tcpdump* was compiled with cryptography enabled. *secret* the ascii text for ESP secret key. We cannot take arbitrary binary value at this moment. The option assumes RFC2406 ESP, not RFC1827 ESP. The option is only for debugging purposes, and the use of this option with truly 'secret' key is discouraged. By presenting IPsec secret key onto command line you make it visible to others, via *ps*(1) and other occasions.
- f** Print 'foreign' internet addresses numerically rather than symbolically (this option is intended to get around serious brain damage in Sun's yp server — usually it hangs forever translating non-local internet numbers).
- F** Use *file* as input for the filter expression. An additional expression given on the command line is ignored.
- i** Listen on *interface*. If unspecified, *tcpdump* searches the system interface list for the lowest numbered, configured up interface (excluding loopback). Ties are broken by choosing the earliest match.

On Linux systems with 2.2 or later kernels, an *interface* argument of "any" can be used to capture packets from all interfaces. Note that captures on the "any" device will not be done in promiscuous mode.
- l** Make stdout line buffered. Useful if you want to see the data while capturing it. E.g., "tcpdump -l | tee dat" or "tcpdump -l > dat & tail -f dat".
- m** Load SMI MIB module definitions from file *module*. This option can be used several times to load several MIB modules into *tcpdump*.
- n** Don't convert addresses (i.e., host addresses, port numbers, etc.) to names.
- N** Don't print domain name qualification of host names. E.g., if you give this flag then *tcpdump* will print "nic" instead of "nic.ddn.mil".
- O** Do not run the packet-matching code optimizer. This is useful only if you suspect a bug in the optimizer.

- p** *Don't* put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, '-p' cannot be used as an abbreviation for 'ether host {local-hw-addr} or ether broadcast'.
- q** Quick (quiet?) output. Print less protocol information so output lines are shorter.
- R** Assume ESP/AH packets to be based on old specification (RFC1825 to RFC1829). If specified, *tcpdump* will not print replay prevention field. Since there is no protocol version field in ESP/AH specification, *tcpdump* cannot deduce the version of ESP/AH protocol.
- r** Read packets from *file* (which was created with the -w option). Standard input is used if *file* is "-".
- S** Print absolute, rather than relative, TCP sequence numbers.
- s** Snarf *snaplen* bytes of data from each packet rather than the default of 68 (with SunOS's NIT, the minimum is actually 96). 68 bytes is adequate for IP, ICMP, TCP and UDP but may truncate protocol information from name server and NFS packets (see below). Packets truncated because of a limited snapshot are indicated in the output with "[*proto*]", where *proto* is the name of the protocol level at which the truncation has occurred. Note that taking larger snapshots both increases the amount of time it takes to process packets and, effectively, decreases the amount of packet buffering. This may cause packets to be lost. You should limit *snaplen* to the smallest number that will capture the protocol information you're interested in. Setting *snaplen* to 0 means use the required length to catch whole packets.
- T** Force packets selected by "*expression*" to be interpreted the specified *type*. Currently known types are **cnfp** (Cisco NetFlow protocol), **rpc** (Remote Procedure Call), **rtp** (Real-Time Applications protocol), **rtcp** (Real-Time Applications control protocol), **snmp** (Simple Network Management Protocol), **vat** (Visual Audio Tool), and **wb** (distributed White Board).
- t** *Don't* print a timestamp on each dump line.
- tt** Print an unformatted timestamp on each dump line.
- ttt** Print a delta (in micro-seconds) between current and previous line on each dump line.
- tttt** Print a timestamp in default format proceeded by date on each dump line.
- u** Print undecoded NFS handles.
- v** (Slightly more) verbose output. For example, the time to live, identification, total length and options in an IP packet are printed. Also enables additional packet integrity checks such as verifying the IP and ICMP header checksum.
- vv** Even more verbose output. For example, additional fields are printed from NFS reply packets, and SMB packets are fully decoded.
- vvv** Even more verbose output. For example, telnet **SB** ... **SE** options are printed in full. With **-X** telnet options are printed in hex as well.
- w** Write the raw packets to *file* rather than parsing and printing them out. They can later be printed with the -r option. Standard output is used if *file* is "-".
- x** Print each packet (minus its link level header) in hex. The smaller of the entire packet or *snaplen* bytes will be printed. Note that this is the entire link-layer packet, so for link layers that pad (e.g. Ethernet), the padding bytes will also be printed when the higher layer packet is shorter than the required padding.
- X** When printing hex, print ascii too. Thus if **-x** is also set, the packet is printed in hex/ascii. This is very handy for analysing new protocols. Even if **-x** is not also set, some parts of some packets may be printed in hex/ascii.

expression

selects which packets will be dumped. If no *expression* is given, all packets on the net will be dumped. Otherwise, only packets for which *expression* is 'true' will be dumped.

The *expression* consists of one or more *primitives*. Primitives usually consist of an *id* (name or number) preceded by one or more qualifiers. There are three different kinds of qualifier:

- type* qualifiers say what kind of thing the *id* name or number refers to. Possible types are **host**, **net** and **port**. E.g., 'host foo', 'net 128.3', 'port 20'. If there is no type qualifier, **host** is assumed.
- dir* qualifiers specify a particular transfer direction to and/or from *id*. Possible directions are **src**, **dst**, **src or dst** and **src and dst**. E.g., 'src foo', 'dst net 128.3', 'src or dst port ftp-data'. If there is no *dir* qualifier, **src or dst** is assumed. For 'null' link layers (i.e. point to point protocols such as slip) the **inbound** and **outbound** qualifiers can be used to specify a desired direction.
- proto* qualifiers restrict the match to a particular protocol. Possible protos are: **ether**, **fddi**, **tr**, **ip**, **ip6**, **arp**, **rarp**, **decnet**, **tcp** and **udp**. E.g., 'ether src foo', 'arp net 128.3', 'tcp port 21'. If there is no *proto* qualifier, all protocols consistent with the type are assumed. E.g., 'src foo' means '(ip or arp or rarp) src foo' (except the latter is not legal syntax), 'net bar' means '(ip or arp or rarp) net bar' and 'port 53' means '(tcp or udp) port 53'.

['fddi' is actually an alias for 'ether'; the parser treats them identically as meaning "the data link level used on the specified network interface." FDDI headers contain Ethernet-like source and destination addresses, and often contain Ethernet-like packet types, so you can filter on these FDDI fields just as with the analogous Ethernet fields. FDDI headers also contain other fields, but you cannot name them explicitly in a filter expression.

Similarly, 'tr' is an alias for 'ether'; the previous paragraph's statements about FDDI headers also apply to Token Ring headers.]

In addition to the above, there are some special 'primitive' keywords that don't follow the pattern: **gateway**, **broadcast**, **less**, **greater** and arithmetic expressions. All of these are described below.

More complex filter expressions are built up by using the words **and**, **or** and **not** to combine primitives. E.g., 'host foo and not port ftp and not port ftp-data'. To save typing, identical qualifier lists can be omitted. E.g., 'tcp dst port ftp or ftp-data or domain' is exactly the same as 'tcp dst port ftp or tcp dst port ftp-data or tcp dst port domain'.

Allowable primitives are:

dst host *host*

True if the IPv4/v6 destination field of the packet is *host*, which may be either an address or a name.

src host *host*

True if the IPv4/v6 source field of the packet is *host*.

host *host*

True if either the IPv4/v6 source or destination of the packet is *host*. Any of the above host expressions can be prepended with the keywords, **ip**, **arp**, **rarp**, or **ip6** as in:

ip host *host*

which is equivalent to:

ether proto *ip* and **host** *host*

If *host* is a name with multiple IP addresses, each address will be checked for a match.

ether dst *ehost*

True if the ethernet destination address is *ehost*. *Ehost* may be either a name from /etc/ethers or a number (see *ethers*(3N) for numeric format).

ether src *ehost*

True if the ethernet source address is *ehost*.

ether host *ehost*

True if either the ethernet source or destination address is *ehost*.

gateway *host*

True if the packet used *host* as a gateway. I.e., the ethernet source or destination address was *host* but neither the IP source nor the IP destination was *host*. *Host* must be a name and must be found both by the machine's host-name-to-IP-address resolution mechanisms (host name file, DNS, NIS, etc.) and by the machine's host-name-to-Ethernet-address resolution mechanism (/etc/ethers, etc.). (An equivalent expression is

ether host *ehost* and not host *host*

which can be used with either names or numbers for *host* / *ehost*.) This syntax does not work in IPv6-enabled configuration at this moment.

dst net *net*

True if the IPv4/v6 destination address of the packet has a network number of *net*. *Net* may be either a name from /etc/networks or a network number (see *networks(4)* for details).

src net *net*

True if the IPv4/v6 source address of the packet has a network number of *net*.

net net True if either the IPv4/v6 source or destination address of the packet has a network number of *net*.

net net mask *netmask*

True if the IP address matches *net* with the specific *netmask*. May be qualified with **src** or **dst**. Note that this syntax is not valid for IPv6 *net*.

net netlen

True if the IPv4/v6 address matches *net* with a netmask *len* bits wide. May be qualified with **src** or **dst**.

dst port *port*

True if the packet is ip/tcp, ip/udp, ip6/tcp or ip6/udp and has a destination port value of *port*. The *port* can be a number or a name used in /etc/services (see *tcp(4P)* and *udp(4P)*). If a name is used, both the port number and protocol are checked. If a number or ambiguous name is used, only the port number is checked (e.g., **dst port 513** will print both tcp/login traffic and udp/who traffic, and **port domain** will print both tcp/domain and udp/domain traffic).

src port *port*

True if the packet has a source port value of *port*.

port *port*

True if either the source or destination port of the packet is *port*. Any of the above port expressions can be prepended with the keywords, **tcp** or **udp**, as in:

tcp src port *port*

which matches only tcp packets whose source port is *port*.

less *length*

True if the packet has a length less than or equal to *length*. This is equivalent to:

len <= length.

greater *length*

True if the packet has a length greater than or equal to *length*. This is equivalent to:

len >= length.

ip proto *protocol*

True if the packet is an IP packet (see *ip(4P)*) of protocol type *protocol*. *Protocol* can be a number or one of the names *icmp*, *icmp6*, *igmp*, *igrp*, *pim*, *ah*, *esp*, *vrp*, *udp*, or *tcp*. Note that the identifiers *tcp*, *udp*, and *icmp* are also keywords and must be escaped via backslash (\), which is \ in the C-shell. Note that this primitive does not chase the protocol header chain.

ip6 proto *protocol*

True if the packet is an IPv6 packet of protocol type *protocol*. Note that this primitive does not chase the protocol header chain.

ip6 protochain *protocol*

True if the packet is IPv6 packet, and contains protocol header with type *protocol* in its protocol header chain. For example,

ip6 protochain 6

matches any IPv6 packet with TCP protocol header in the protocol header chain. The packet may contain, for example, authentication header, routing header, or hop-by-hop option header, between IPv6 header and TCP header. The BPF code emitted by this primitive is complex and cannot be optimized by BPF optimizer code in *tcpdump*, so this can be somewhat slow.

ip protochain *protocol*

Equivalent to **ip6 protochain *protocol***, but this is for IPv4.

ether broadcast

True if the packet is an ethernet broadcast packet. The *ether* keyword is optional.

ip broadcast

True if the packet is an IP broadcast packet. It checks for both the all-zeroes and all-ones broadcast conventions, and looks up the local subnet mask.

ether multicast

True if the packet is an ethernet multicast packet. The *ether* keyword is optional. This is shorthand for '**ether[0] & 1 != 0**'.

ip multicast

True if the packet is an IP multicast packet.

ip6 multicast

True if the packet is an IPv6 multicast packet.

ether proto *protocol*

True if the packet is of ether type *protocol*. *Protocol* can be a number or one of the names *ip*, *ip6*, *arp*, *rarp*, *atalk*, *aarp*, *decnet*, *sca*, *lat*, *mopdl*, *moprc*, *iso*, *stp*, *ipx*, or *netbeui*. Note these identifiers are also keywords and must be escaped via backslash (\).

[In the case of FDDI (e.g., '**fdi protocol arp**') and Token Ring (e.g., '**tr protocol arp**'), for most of those protocols, the protocol identification comes from the 802.2 Logical Link Control (LLC) header, which is usually layered on top of the FDDI or Token Ring header.

When filtering for most protocol identifiers on FDDI or Token Ring, *tcpdump* checks only the protocol ID field of an LLC header in so-called SNAP format with an Organizational Unit Identifier (OUI) of 0x000000, for encapsulated Ethernet; it doesn't check whether the packet is in SNAP format with an OUI of 0x000000.

The exceptions are *iso*, for which it checks the DSAP (Destination Service Access Point) and SSAP (Source Service Access Point) fields of the LLC header, *stp* and *netbeui*, where it checks the DSAP of the LLC header, and *atalk*, where it checks for a SNAP-format packet with an OUI of 0x080007 and the Appletalk etype.

In the case of Ethernet, *tcpdump* checks the Ethernet type field for most of those protocols; the exceptions are *iso*, *sap*, and *netbeui*, for which it checks for an 802.3 frame and then checks the LLC header as it does for FDDI and Token Ring, *atalk*, where it checks both for the Appletalk etype in an Ethernet frame and for a SNAP-format packet as it does for FDDI and Token Ring, *aarp*, where it checks for the Appletalk ARP etype in either an Ethernet frame or an 802.2 SNAP frame with an OUI of 0x000000, and *ipx*, where it checks for the IPX etype in an Ethernet frame, the IPX DSAP in the LLC header, the 802.3 with no LLC header encapsulation of IPX, and the IPX etype in a SNAP

frame.]

decnet src *host*

True if the DECNET source address is *host*, which may be an address of the form “10.123”, or a DECNET host name. [DECNET host name support is only available on Ultrix systems that are configured to run DECNET.]

decnet dst *host*

True if the DECNET destination address is *host*.

decnet host *host*

True if either the DECNET source or destination address is *host*.

ip, ip6, arp, rarp, atalk, aarp, decnet, iso, stp, ipx, netbeui

Abbreviations for:

ether proto *p*

where *p* is one of the above protocols.

lat, moprc, mopdl

Abbreviations for:

ether proto *p*

where *p* is one of the above protocols. Note that *tcpdump* does not currently know how to parse these protocols.

vlan [*vlan_id*]

True if the packet is an IEEE 802.1Q VLAN packet. If [*vlan_id*] is specified, only true is the packet has the specified *vlan_id*. Note that the first **vlan** keyword encountered in *expression* changes the decoding offsets for the remainder of *expression* on the assumption that the packet is a VLAN packet.

tcp, udp, icmp

Abbreviations for:

ip proto *p* or **ip6 proto** *p*

where *p* is one of the above protocols.

iso proto *protocol*

True if the packet is an OSI packet of protocol type *protocol*. *Protocol* can be a number or one of the names *clnp*, *esis*, or *isis*.

clnp, esis, isis

Abbreviations for:

iso proto *p*

where *p* is one of the above protocols. Note that *tcpdump* does an incomplete job of parsing these protocols.

expr relop expr

True if the relation holds, where *relop* is one of **>**, **<**, **>=**, **<=**, **=**, **!=**, and *expr* is an arithmetic expression composed of integer constants (expressed in standard C syntax), the normal binary operators **+**, **-**, *****, **/**, **&**, **[]**, a length operator, and special packet data accessors. To access data inside the packet, use the following syntax:

proto [*expr* : *size*]

Proto is one of **ether**, **fddi**, **tr**, **ppp**, **slip**, **link**, **ip**, **arp**, **rarp**, **tcp**, **udp**, **icmp** or **ip6**, and indicates the protocol layer for the index operation. (**ether**, **fddi**, **tr**, **ppp**, **slip** and **link** all refer to the link layer.) Note that *tcp*, *udp* and other upper-layer protocol types only apply to IPv4, not IPv6 (this will be fixed in the future). The byte offset, relative to the indicated protocol layer, is given by *expr*. *Size* is optional and indicates the number of bytes in the field of interest; it can be either one, two, or four, and defaults to one. The length operator, indicated by the keyword **len**, gives the length of the packet.

For example, **‘ether[0] & 1 != 0’** catches all multicast traffic. The expression **‘ip[0] &**

0xf != 5' catches all IP packets with options. The expression **'ip[6:2] & 0x1fff = 0'** catches only unfragmented datagrams and frag zero of fragmented datagrams. This check is implicitly applied to the **tcp** and **udp** index operations. For instance, **tcp[0]** always means the first byte of the TCP *header*, and never means the first byte of an intervening fragment.

Some offsets and field values may be expressed as names rather than as numeric values. The following protocol header field offsets are available: **icmptype** (ICMP type field), **icmpcode** (ICMP code field), and **tcpflags** (TCP flags field).

The following ICMP type field values are available: **icmp-echoreply**, **icmp-unreach**, **icmp-sourcequench**, **icmp-redirect**, **icmp-echo**, **icmp-routeradvert**, **icmp-routersolicit**, **icmp-timxceed**, **icmp-paramprob**, **icmp-tstamp**, **icmp-tstampreply**, **icmp-ireq**, **icmp-ireqreply**, **icmp-maskreq**, **icmp-maskreply**.

The following TCP flags field values are available: **tcp-fin**, **tcp-syn**, **tcp-rst**, **tcp-push**, **tcp-push**, **tcp-ack**, **tcp-urg**.

Primitives may be combined using:

A parenthesized group of primitives and operators (parentheses are special to the Shell and must be escaped).

Negation ('!' or **'not'**).

Concatenation ('&&' or **'and'**).

Alternation ('||' or **'or'**).

Negation has highest precedence. Alternation and concatenation have equal precedence and associate left to right. Note that explicit **and** tokens, not juxtaposition, are now required for concatenation.

If an identifier is given without a keyword, the most recent keyword is assumed. For example,

not host vs and ace

is short for

not host vs and host ace

which should not be confused with

not (host vs or ace)

Expression arguments can be passed to *tcpdump* as either a single argument or as multiple arguments, whichever is more convenient. Generally, if the expression contains Shell metacharacters, it is easier to pass it as a single, quoted argument. Multiple arguments are concatenated with spaces before being parsed.

EXAMPLES

To print all packets arriving at or departing from *sundown*:

tcpdump host sundown

To print traffic between *helios* and either *hot* or *ace*:

tcpdump host helios and \(hot or ace \)

To print all IP packets between *ace* and any host except *helios*:

tcpdump ip host ace and not helios

To print all traffic between local hosts and hosts at Berkeley:

tcpdump net ucb-ether

To print all ftp traffic through internet gateway *snup*: (note that the expression is quoted to prevent the shell from (mis-)interpreting the parentheses):

tcpdump 'gateway snup and (port ftp or ftp-data)'

To print traffic neither sourced from nor destined for local hosts (if you gateway to one other net, this stuff

should never make it onto your local net).

tcpdump ip and not net localnet

To print the start and end packets (the SYN and FIN packets) of each TCP conversation that involves a non-local host.

tcpdump 'tcp[tcpflags] & (tcp-syn|tcp-fin) != 0 and not src and dst net localnet'

To print IP packets longer than 576 bytes sent through gateway *snup*:

tcpdump 'gateway snup and ip[2:2] > 576'

To print IP broadcast or multicast packets that were *not* sent via ethernet broadcast or multicast:

tcpdump 'ether[0] & 1 = 0 and ip[16] >= 224'

To print all ICMP packets that are not echo requests/replies (i.e., not ping packets):

tcpdump 'icmp[icmptype] != icmp-echo and icmp[icmptype] != icmp-echoreply'

OUTPUT FORMAT

The output of *tcpdump* is protocol dependent. The following gives a brief description and examples of most of the formats.

Link Level Headers

If the '-e' option is given, the link level header is printed out. On ethernet, the source and destination addresses, protocol, and packet length are printed.

On FDDI networks, the '-e' option causes *tcpdump* to print the 'frame control' field, the source and destination addresses, and the packet length. (The 'frame control' field governs the interpretation of the rest of the packet. Normal packets (such as those containing IP datagrams) are 'async' packets, with a priority value between 0 and 7; for example, 'async4'. Such packets are assumed to contain an 802.2 Logical Link Control (LLC) packet; the LLC header is printed if it is *not* an ISO datagram or a so-called SNAP packet.

On Token Ring networks, the '-e' option causes *tcpdump* to print the 'access control' and 'frame control' fields, the source and destination addresses, and the packet length. As on FDDI networks, packets are assumed to contain an LLC packet. Regardless of whether the '-e' option is specified or not, the source routing information is printed for source-routed packets.

(N.B.: The following description assumes familiarity with the SLIP compression algorithm described in RFC-1144.)

On SLIP links, a direction indicator ("I" for inbound, "O" for outbound), packet type, and compression information are printed out. The packet type is printed first. The three types are *ip*, *utcp*, and *ctcp*. No further link information is printed for *ip* packets. For TCP packets, the connection identifier is printed following the type. If the packet is compressed, its encoded header is printed out. The special cases are printed out as *S+n and *SA+n, where *n* is the amount by which the sequence number (or sequence number and ack) has changed. If it is not a special case, zero or more changes are printed. A change is indicated by U (urgent pointer), W (window), A (ack), S (sequence number), and I (packet ID), followed by a delta (+n or -n), or a new value (=n). Finally, the amount of data in the packet and compressed header length are printed.

For example, the following line shows an outbound compressed TCP packet, with an implicit connection identifier; the ack has changed by 6, the sequence number by 49, and the packet ID by 6; there are 3 bytes of data and 6 bytes of compressed header:

O ctcp * A+6 S+49 I+6 3 (6)

ARP/RARP Packets

Arp/rarp output shows the type of request and its arguments. The format is intended to be self explanatory. Here is a short sample taken from the start of an 'rlogin' from host *rtsg* to host *csam*:

```
arp who-has csam tell rtsg
arp reply csam is-at CSAM
```

The first line says that *rtsg* sent an arp packet asking for the ethernet address of internet host *csam*. *Csam* replies with its ethernet address (in this example, ethernet addresses are in caps and internet addresses in lower case).

This would look less redundant if we had done *tcpdump -n*:

```
arp who-has 128.3.254.6 tell 128.3.254.68
arp reply 128.3.254.6 is-at 02:07:01:00:01:c4
```

If we had done *tcpdump -e*, the fact that the first packet is broadcast and the second is point-to-point would be visible:

```
RTSG Broadcast 0806 64: arp who-has csam tell rtsg
CSAM RTSG 0806 64: arp reply csam is-at CSAM
```

For the first packet this says the ethernet source address is *RTSG*, the destination is the ethernet broadcast address, the type field contained hex 0806 (type *ETHER_ARP*) and the total length was 64 bytes.

TCP Packets

(*N.B.:The following description assumes familiarity with the TCP protocol described in RFC-793. If you are not familiar with the protocol, neither this description nor tcpdump will be of much use to you.*)

The general format of a tcp protocol line is:

```
src > dst: flags data-seqno ack window urgent options
```

Src and *dst* are the source and destination IP addresses and ports. *Flags* are some combination of S (SYN), F (FIN), P (PUSH) or R (RST) or a single '.' (no flags). *Data-seqno* describes the portion of sequence space covered by the data in this packet (see example below). *Ack* is sequence number of the next data expected the other direction on this connection. *Window* is the number of bytes of receive buffer space available the other direction on this connection. *Urg* indicates there is 'urgent' data in the packet. *Options* are tcp options enclosed in angle brackets (e.g., <mss 1024>).

Src, *dst* and *flags* are always present. The other fields depend on the contents of the packet's tcp protocol header and are output only if appropriate.

Here is the opening portion of an rlogin from host *rtsg* to host *csam*.

```
rtsg.1023 > csam.login: S 768512:768512(0) win 4096 <mss 1024>
csam.login > rtsg.1023: S 947648:947648(0) ack 768513 win 4096 <mss 1024>
rtsg.1023 > csam.login: . ack 1 win 4096
rtsg.1023 > csam.login: P 1:2(1) ack 1 win 4096
csam.login > rtsg.1023: . ack 2 win 4096
rtsg.1023 > csam.login: P 2:21(19) ack 1 win 4096
csam.login > rtsg.1023: P 1:2(1) ack 21 win 4077
csam.login > rtsg.1023: P 2:3(1) ack 21 win 4077 urg 1
csam.login > rtsg.1023: P 3:4(1) ack 21 win 4077 urg 1
```

The first line says that tcp port 1023 on *rtsg* sent a packet to port *login* on *csam*. The **S** indicates that the *SYN* flag was set. The packet sequence number was 768512 and it contained no data. (The notation is 'first:last(nbytes)' which means 'sequence numbers *first* up to but not including *last* which is *nbytes* bytes of user data'.) There was no piggy-backed ack, the available receive window was 4096 bytes and there was a max-segment-size option requesting an mss of 1024 bytes.

Csam replies with a similar packet except it includes a piggy-backed ack for *rtsg*'s SYN. *Rtsg* then acks *csam*'s SYN. The '.' means no flags were set. The packet contained no data so there is no data sequence number. Note that the ack sequence number is a small integer (1). The first time *tcpdump* sees a tcp 'conversation', it prints the sequence number from the packet. On subsequent packets of the conversation, the difference between the current packet's sequence number and this initial sequence number is printed. This means that sequence numbers after the first can be interpreted as relative byte positions in the conversation's data stream (with the first data byte each direction being '1'). '-S' will override this feature, causing

the original sequence numbers to be output.

On the 6th line, rtsg sends csam 19 bytes of data (bytes 2 through 20 in the rtsg → csam side of the conversation). The PUSH flag is set in the packet. On the 7th line, csam says it's received data sent by rtsg up to but not including byte 21. Most of this data is apparently sitting in the socket buffer since csam's receive window has gotten 19 bytes smaller. Csam also sends one byte of data to rtsg in this packet. On the 8th and 9th lines, csam sends two bytes of urgent, pushed data to rtsg.

If the snapshot was small enough that *tcpdump* didn't capture the full TCP header, it interprets as much of the header as it can and then reports "[*tcp*]" to indicate the remainder could not be interpreted. If the header contains a bogus option (one with a length that's either too small or beyond the end of the header), *tcpdump* reports it as "[*bad opt*]" and does not interpret any further options (since it's impossible to tell where they start). If the header length indicates options are present but the IP datagram length is not long enough for the options to actually be there, *tcpdump* reports it as "[*bad hdr length*]".

Capturing TCP packets with particular flag combinations (SYN-ACK, URG-ACK, etc.)

There are 8 bits in the control bits section of the TCP header:

CWR | *ECE* | *URG* | *ACK* | *PSH* | *RST* | *SYN* | *FIN*

Let's assume that we want to watch packets used in establishing a TCP connection. Recall that TCP uses a 3-way handshake protocol when it initializes a new connection; the connection sequence with regard to the TCP control bits is

- 1) Caller sends SYN
- 2) Recipient responds with SYN, ACK
- 3) Caller sends ACK

Now we're interested in capturing packets that have only the SYN bit set (Step 1). Note that we don't want packets from step 2 (SYN-ACK), just a plain initial SYN. What we need is a correct filter expression for *tcpdump*.

Recall the structure of a TCP header without options:

| | | |
|-----------------------|------------------|-------------------------------|
| 0 | 15 | 31 |
| ----- | | |
| source port | destination port | |
| ----- | | |
| sequence number | | |
| ----- | | |
| acknowledgment number | | |
| ----- | | |
| HL | rsvd | C E U A P R S F window size |
| ----- | | |
| TCP checksum | | urgent pointer |
| ----- | | |

A TCP header usually holds 20 octets of data, unless options are present. The first line of the graph contains octets 0 - 3, the second line shows octets 4 - 7 etc.

Starting to count with 0, the relevant TCP control bits are contained in octet 13:

| | | | | |
|------------|------|-----------------|-------------|----|
| 0 | 7 | 15 | 23 | 31 |
| ----- | | | | |
| HL | rsvd | C E U A P R S F | window size | |
| ----- | | | | |
| 13th octet | | | | |

Let's have a closer look at octet no. 13:

|


```

|-----|
|C|E|U|A|P|R|S|F|
|-----|
|7 5 3 0|

```

These are the TCP control bits we are interested in. We have numbered the bits in this octet from 0 to 7, right to left, so the PSH bit is bit number 3, while the URG bit is number 5.

Recall that we want to capture packets with only SYN set. Let's see what happens to octet 13 if a TCP datagram arrives with the SYN bit set in its header:

```

|C|E|U|A|P|R|S|F|
|-----|
|0 0 0 0 0 1 0|
|-----|
|7 6 5 4 3 2 1 0|

```

Looking at the control bits section we see that only bit number 1 (SYN) is set.

Assuming that octet number 13 is an 8-bit unsigned integer in network byte order, the binary value of this octet is

00000010

and its decimal representation is

```

 7 6 5 4 3 2 1 0
0*2 + 0*2 + 0*2 + 0*2 + 0*2 + 0*2 + 1*2 + 0*2 = 2

```

We're almost done, because now we know that if only SYN is set, the value of the 13th octet in the TCP header, when interpreted as a 8-bit unsigned integer in network byte order, must be exactly 2.

This relationship can be expressed as

tcp[13] == 2

We can use this expression as the filter for *tcpdump* in order to watch packets which have only SYN set:

tcpdump -i xl0 tcp[13] == 2

The expression says "let the 13th octet of a TCP datagram have the decimal value 2", which is exactly what we want.

Now, let's assume that we need to capture SYN packets, but we don't care if ACK or any other TCP control bit is set at the same time. Let's see what happens to octet 13 when a TCP datagram with SYN-ACK set arrives:

```

|C|E|U|A|P|R|S|F|
|-----|
|0 0 0 1 0 0 1 0|
|-----|
|7 6 5 4 3 2 1 0|

```

Now bits 1 and 4 are set in the 13th octet. The binary value of octet 13 is

00010010

which translates to decimal

```

 7 6 5 4 3 2 1 0
0*2 + 0*2 + 0*2 + 1*2 + 0*2 + 0*2 + 1*2 + 0*2 = 18

```

Now we can't just use 'tcp[13] == 18' in the *tcpdump* filter expression, because that would select only those packets that have SYN-ACK set, but not those with only SYN set. Remember that we don't care if ACK or any other control bit is set as long as SYN is set.

In order to achieve our goal, we need to logically AND the binary value of octet 13 with some other value to preserve the SYN bit. We know that we want SYN to be set in any case, so we'll logically AND the

value in the 13th octet with the binary value of a SYN:

```

      00010010 SYN-ACK      00000010 SYN
      AND 00000010 (we want SYN) AND 00000010 (we want SYN)
      -----
      = 00000010          = 00000010

```

We see that this AND operation delivers the same result regardless whether ACK or another TCP control bit is set. The decimal representation of the AND value as well as the result of this operation is 2 (binary 00000010), so we know that for packets with SYN set the following relation must hold true:

$$((\text{value of octet 13}) \text{ AND } (2)) == (2)$$

This points us to the *tcpdump* filter expression

```
tcpdump -i xl0 'tcp[13] & 2 == 2'
```

Note that you should use single quotes or a backslash in the expression to hide the AND ('&') special character from the shell.

UDP Packets

UDP format is illustrated by this *rwho* packet:

```
actinide.who > broadcast.who: udp 84
```

This says that port *who* on host *actinide* sent a udp datagram to port *who* on host *broadcast*, the Internet broadcast address. The packet contained 84 bytes of user data.

Some UDP services are recognized (from the source or destination port number) and the higher level protocol information printed. In particular, Domain Name service requests (RFC-1034/1035) and Sun RPC calls (RFC-1050) to NFS.

UDP Name Server Requests

(N.B.:The following description assumes familiarity with the Domain Service protocol described in RFC-1035. If you are not familiar with the protocol, the following description will appear to be written in greek.)

Name server requests are formatted as

```
src > dst: id op? flags qtype qclass name (len)

h2opolo.1538 > helios.domain: 3+ A? ucbvax.berkeley.edu. (37)
```

Host *h2opolo* asked the domain server on *helios* for an address record (qtype=A) associated with the name *ucbvax.berkeley.edu*. The query id was '3'. The '+' indicates the *recursion desired* flag was set. The query length was 37 bytes, not including the UDP and IP protocol headers. The query operation was the normal one, *Query*, so the op field was omitted. If the op had been anything else, it would have been printed between the '3' and the '+'. Similarly, the qclass was the normal one, *C_IN*, and omitted. Any other qclass would have been printed immediately after the 'A'.

A few anomalies are checked and may result in extra fields enclosed in square brackets: If a query contains an answer, authority records or additional records section, *ancount*, *nscount*, or *arcount* are printed as '[na]', '[nm]' or '[nau]' where *n* is the appropriate count. If any of the response bits are set (AA, RA or rcode) or any of the 'must be zero' bits are set in bytes two and three, '[b2&3=x]' is printed, where *x* is the hex value of header bytes two and three.

UDP Name Server Responses

Name server responses are formatted as

```
src > dst: id op rcode flags a/n/au type class data (len)
```

```
helios.domain > h2opolo.1538: 3 3/3/7 A 128.32.137.3 (273)
helios.domain > h2opolo.1537: 2 NXDomain* 0/1/0 (97)
```

In the first example, *helios* responds to query id 3 from *h2opolo* with 3 answer records, 3 name server records and 7 additional records. The first answer record is type A (address) and its data is internet address 128.32.137.3. The total size of the response was 273 bytes, excluding UDP and IP headers. The op (Query) and response code (NoError) were omitted, as was the class (C_IN) of the A record.

In the second example, *helios* responds to query 2 with a response code of non-existent domain (NXDomain) with no answers, one name server and no authority records. The '*' indicates that the *authoritative answer* bit was set. Since there were no answers, no type, class or data were printed.

Other flag characters that might appear are '-' (recursion available, RA, *not* set) and '|' (truncated message, TC, set). If the 'question' section doesn't contain exactly one entry, '[nq]' is printed.

Note that name server requests and responses tend to be large and the default *snaplen* of 68 bytes may not capture enough of the packet to print. Use the *-s* flag to increase the *snaplen* if you need to seriously investigate name server traffic. '*-s 128*' has worked well for me.

SMB/CIFS decoding

tcpdump now includes fairly extensive SMB/CIFS/NBT decoding for data on UDP/137, UDP/138 and TCP/139. Some primitive decoding of IPX and NetBEUI SMB data is also done.

By default a fairly minimal decode is done, with a much more detailed decode done if *-v* is used. Be warned that with *-v* a single SMB packet may take up a page or more, so only use *-v* if you really want all the gory details.

If you are decoding SMB sessions containing unicode strings then you may wish to set the environment variable *USE_UNICODE* to 1. A patch to auto-detect unicode strings would be welcome.

For information on SMB packet formats and what all the fields mean see www.cifs.org or the [pub/samba/specs/](http://pub.samba/specs/) directory on your favourite samba.org mirror site. The SMB patches were written by Andrew Tridgell (tridge@samba.org).

NFS Requests and Replies

Sun NFS (Network File System) requests and replies are printed as:

```
src.xid > dst.nfs: len op args
src.nfs > dst.xid: reply stat len op results

sushi.6709 > wr1.nfs: 112 readlink fh 21,24/10.73165
wr1.nfs > sushi.6709: reply ok 40 readlink "../var"
sushi.201b > wr1.nfs:
    144 lookup fh 9,74/4096.6878 "xcolors"
wr1.nfs > sushi.201b:
    reply ok 128 lookup fh 9,74/4134.3150
```

In the first line, host *sushi* sends a transaction with id 6709 to *wr1* (note that the number following the *src* host is a transaction id, *not* the source port). The request was 112 bytes, excluding the UDP and IP headers. The operation was a *readlink* (read symbolic link) on file handle (*fh*) 21,24/10.731657119. (If one is lucky, as in this case, the file handle can be interpreted as a major,minor device number pair, followed by the inode number and generation number.) *Wr1* replies 'ok' with the contents of the link.

In the third line, *sushi* asks *wr1* to lookup the name '*xcolors*' in directory file 9,74/4096.6878. Note that the data printed depends on the operation type. The format is intended to be self explanatory if read in

conjunction with an NFS protocol spec.

If the `-v` (verbose) flag is given, additional information is printed. For example:

```
sushi.1372a > wr1.nfs:
      148 read fh 21,11/12.195 8192 bytes @ 24576
wr1.nfs > sushi.1372a:
      reply ok 1472 read REG 100664 ids 417/0 sz 29388
```

(`-v` also prints the IP header TTL, ID, length, and fragmentation fields, which have been omitted from this example.) In the first line, *sushi* asks *wrl* to read 8192 bytes from file 21,11/12.195, at byte offset 24576. *Wrl* replies 'ok'; the packet shown on the second line is the first fragment of the reply, and hence is only 1472 bytes long (the other bytes will follow in subsequent fragments, but these fragments do not have NFS or even UDP headers and so might not be printed, depending on the filter expression used). Because the `-v` flag is given, some of the file attributes (which are returned in addition to the file data) are printed: the file type ("REG", for regular file), the file mode (in octal), the uid and gid, and the file size.

If the `-v` flag is given more than once, even more details are printed.

Note that NFS requests are very large and much of the detail won't be printed unless *snapplen* is increased. Try using '`-s 192`' to watch NFS traffic.

NFS reply packets do not explicitly identify the RPC operation. Instead, *tcpdump* keeps track of "recent" requests, and matches them to the replies using the transaction ID. If a reply does not closely follow the corresponding request, it might not be parsable.

AFS Requests and Replies

Transarc AFS (Andrew File System) requests and replies are printed as:

```
src.sport > dst.dport: rx packet-type
src.sport > dst.dport: rx packet-type service call call-name args
src.sport > dst.dport: rx packet-type service reply call-name args

elvis.7001 > pike.afsfs:
      rx data fs call rename old fid 536876964/1/1 ".newsrsrc.new"
      new fid 536876964/1/1 ".newsrsrc"
pike.afsfs > elvis.7001: rx data fs reply rename
```

In the first line, host elvis sends a RX packet to pike. This was a RX data packet to the fs (fileserver) service, and is the start of an RPC call. The RPC call was a rename, with the old directory file id of 536876964/1/1 and an old filename of '.newsrsrc.new', and a new directory file id of 536876964/1/1 and a new filename of '.newsrsrc'. The host pike responds with a RPC reply to the rename call (which was successful, because it was a data packet and not an abort packet).

In general, all AFS RPCs are decoded at least by RPC call name. Most AFS RPCs have at least some of the arguments decoded (generally only the 'interesting' arguments, for some definition of interesting).

The format is intended to be self-describing, but it will probably not be useful to people who are not familiar with the workings of AFS and RX.

If the `-v` (verbose) flag is given twice, acknowledgement packets and additional header information is printed, such as the the RX call ID, call number, sequence number, serial number, and the RX packet flags.

If the `-v` flag is given twice, additional information is printed, such as the the RX call ID, serial number, and the RX packet flags. The MTU negotiation information is also printed from RX ack packets.

If the `-v` flag is given three times, the security index and service id are printed.

Error codes are printed for abort packets, with the exception of Ubik beacon packets (because abort packets are used to signify a yes vote for the Ubik protocol).

Note that AFS requests are very large and many of the arguments won't be printed unless *snaplen* is increased. Try using **'-s 256'** to watch AFS traffic.

AFS reply packets do not explicitly identify the RPC operation. Instead, *tcpdump* keeps track of "recent" requests, and matches them to the replies using the call number and service ID. If a reply does not closely follow the corresponding request, it might not be parsable.

KIP Appletalk (DDP in UDP)

Appletalk DDP packets encapsulated in UDP datagrams are de-encapsulated and dumped as DDP packets (i.e., all the UDP header information is discarded). The file */etc/atalk.names* is used to translate appletalk net and node numbers to names. Lines in this file have the form

```
number name
1.254      ether
16.1       icsd-net
1.254.110  ace
```

The first two lines give the names of appletalk networks. The third line gives the name of a particular host (a host is distinguished from a net by the 3rd octet in the number – a net number *must* have two octets and a host number *must* have three octets.) The number and name should be separated by whitespace (blanks or tabs). The */etc/atalk.names* file may contain blank lines or comment lines (lines starting with a '#').

Appletalk addresses are printed in the form

```
net.host.port
144.1.209.2 > icsd-net.112.220
office.2 > icsd-net.112.220
jssmag.149.235 > icsd-net.2
```

(If the */etc/atalk.names* doesn't exist or doesn't contain an entry for some appletalk host/net number, addresses are printed in numeric form.) In the first example, NBP (DDP port 2) on net 144.1 node 209 is sending to whatever is listening on port 220 of net icsd node 112. The second line is the same except the full name of the source node is known ('office'). The third line is a send from port 235 on net jssmag node 149 to broadcast on the icsd-net NBP port (note that the broadcast address (255) is indicated by a net name with no host number – for this reason it's a good idea to keep node names and net names distinct in */etc/atalk.names*).

NBP (name binding protocol) and ATP (Appletalk transaction protocol) packets have their contents interpreted. Other protocols just dump the protocol name (or number if no name is registered for the protocol) and packet size.

NBP packets are formatted like the following examples:

```
icsd-net.112.220 > jssmag.2: nbp-lkup 190: "=:LaserWriter@*"
jssmag.209.2 > icsd-net.112.220: nbp-reply 190: "RM1140:LaserWriter@*" 250
techpit.2 > icsd-net.112.220: nbp-reply 190: "techpit:LaserWriter@*" 186
```

The first line is a name lookup request for laserwriters sent by net icsd host 112 and broadcast on net jssmag. The nbp id for the lookup is 190. The second line shows a reply for this request (note that it has the same id) from host jssmag.209 saying that it has a laserwriter resource named "RM1140" registered on port 250. The third line is another reply to the same request saying host techpit has laserwriter "techpit" registered on port 186.

ATP packet formatting is demonstrated by the following example:

```
jssmag.209.165 > helios.132: atp-req 12266<0-7> 0xae030001
```

```

helios.132 > jssmag.209.165: atp-resp 12266:0 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:1 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:2 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:3 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:4 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:5 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:6 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp*12266:7 (512) 0xae040000
jssmag.209.165 > helios.132: atp-req 12266<3,5> 0xae030001
helios.132 > jssmag.209.165: atp-resp 12266:3 (512) 0xae040000
helios.132 > jssmag.209.165: atp-resp 12266:5 (512) 0xae040000
jssmag.209.165 > helios.132: atp-rel 12266<0-7> 0xae030001
jssmag.209.133 > helios.132: atp-req* 12267<0-7> 0xae030002

```

Jssmag.209 initiates transaction id 12266 with host helios by requesting up to 8 packets (the '<0-7>'). The hex number at the end of the line is the value of the 'userdata' field in the request.

Helios responds with 8 512-byte packets. The ':digit' following the transaction id gives the packet sequence number in the transaction and the number in parens is the amount of data in the packet, excluding the atp header. The '*' on packet 7 indicates that the EOM bit was set.

Jssmag.209 then requests that packets 3 & 5 be retransmitted. Helios resends them then jssmag.209 releases the transaction. Finally, jssmag.209 initiates the next request. The '*' on the request indicates that XO ('exactly once') was *not* set.

IP Fragmentation

Fragmented Internet datagrams are printed as

```

(frag id:size@offset+)
(frag id:size@offset)

```

(The first form indicates there are more fragments. The second indicates this is the last fragment.)

Id is the fragment id. *Size* is the fragment size (in bytes) excluding the IP header. *Offset* is this fragment's offset (in bytes) in the original datagram.

The fragment information is output for each fragment. The first fragment contains the higher level protocol header and the frag info is printed after the protocol info. Fragments after the first contain no higher level protocol header and the frag info is printed after the source and destination addresses. For example, here is part of an ftp from arizona.edu to lbl-rtsg.arpa over a CSNET connection that doesn't appear to handle 576 byte datagrams:

```

arizona.ftp-data > rtsg.1170: . 1024:1332(308) ack 1 win 4096 (frag 595a:328@0+)
arizona > rtsg: (frag 595a:204@328)
rtsg.1170 > arizona.ftp-data: . ack 1536 win 2560

```

There are a couple of things to note here: First, addresses in the 2nd line don't include port numbers. This is because the TCP protocol information is all in the first fragment and we have no idea what the port or sequence numbers are when we print the later fragments. Second, the tcp sequence information in the first line is printed as if there were 308 bytes of user data when, in fact, there are 512 bytes (308 in the first frag and 204 in the second). If you are looking for holes in the sequence space or trying to match up acks with packets, this can fool you.

A packet with the IP *don't fragment* flag is marked with a trailing **(DF)**.

Timestamps

By default, all output lines are preceded by a timestamp. The timestamp is the current clock time in the

form

hh:mm:ss.frac

and is as accurate as the kernel's clock. The timestamp reflects the time the kernel first saw the packet. No attempt is made to account for the time lag between when the ethernet interface removed the packet from the wire and when the kernel serviced the 'new packet' interrupt.

SEE ALSO

traffic(1C), nit(4P), bpf(4), pcap(3)

AUTHORS

The original authors are:

Van Jacobson, Craig Leres and Steven McCanne, all of the Lawrence Berkeley National Laboratory, University of California, Berkeley, CA.

It is currently being maintained by tcpdump.org.

The current version is available via [http](http://www.tcpdump.org/):

<http://www.tcpdump.org/>

The original distribution is available via anonymous [ftp](ftp://ftp.ee.lbl.gov/tcpdump.tar.Z):

<ftp://ftp.ee.lbl.gov/tcpdump.tar.Z>

IPv6/IPsec support is added by WIDE/KAME project. This program uses Eric Young's SSLeay library, under specific configuration.

BUGS

Please send problems, bugs, questions, desirable enhancements, etc. to:

tcpdump-workers@tcpdump.org

Please send source code contributions, etc. to:

patches@tcpdump.org

NIT doesn't let you watch your own outbound traffic, BPF will. We recommend that you use the latter.

On Linux systems with 2.0[.x] kernels:

packets on the loopback device will be seen twice;

packet filtering cannot be done in the kernel, so that all packets must be copied from the kernel in order to be filtered in user mode;

all of a packet, not just the part that's within the snapshot length, will be copied from the kernel (the 2.0[.x] packet capture mechanism, if asked to copy only part of a packet to userland, will not report the true length of the packet; this would cause most IP packets to get an error from **tcpdump**);

capturing on some PPP devices won't work correctly.

We recommend that you upgrade to a 2.2 or later kernel.

Some attempt should be made to reassemble IP fragments or, at least to compute the right length for the higher level protocol.

Name server inverse queries are not dumped correctly: the (empty) question section is printed rather than real query in the answer section. Some believe that inverse queries are themselves a bug and prefer to fix the program generating them rather than *tcpdump*.

A packet trace that crosses a daylight savings time change will give skewed time stamps (the time change is ignored).

Filter expressions that manipulate FDDI or Token Ring headers assume that all FDDI and Token Ring packets are SNAP-encapsulated Ethernet packets. This is true for IP, ARP, and DECNET Phase IV, but is not true for protocols such as ISO CLNS. Therefore, the filter may inadvertently accept certain packets that do not properly match the filter expression.

Filter expressions on fields other than those that manipulate Token Ring headers will not correctly handle source-routed Token Ring packets.

ip6 proto should chase header chain, but at this moment it does not. **ip6 protochain** is supplied for this behavior.

Arithmetic expression against transport layer headers, like **tcp[0]**, does not work against IPv6 packets. It only looks at IPv4 packets.

NAME

termcap – terminal capability database

DESCRIPTION

The termcap database is an obsolete facility for describing the capabilities of character-cell terminals and printers. It is retained only for capability with old programs; new ones should use the **terminfo**(5) database and associated libraries.

edrc/etc/termcap or */etc/termcap* is an ASCII file (the database master) that lists the capabilities of many different types of terminals. Programs can read termcap to find the particular escape codes needed to control the visual attributes of the terminal actually in use. (Other aspects of the terminal are handled by **stty**(1).) The termcap database is indexed on the **TERM** environment variable.

Termcap entries must be defined on a single logical line, with '\ ' used to suppress the newline. Fields are separated by ':'. The first field of each entry starts at the left-hand margin, and contains a list of names for the terminal, separated by '| '.

The first subfield may (in BSD termcap entries from versions 4.3 and prior) contain a short name consisting of two characters. This short name may consist of capital or small letters. In 4.4BSD termcap entries this field is omitted.

The second subfield (first, in the newer 4.4BSD format) contains the name used by the environment variable **TERM**. It should be spelled in lowercase letters. Selectable hardware capabilities should be marked by appending a hyphen and a suffix to this name. See below for an example. Usual suffixes are w (more than 80 characters wide), am (automatic margins), nam (no automatic margins), and rv (reverse video display). The third subfield contains a long and descriptive name for this termcap entry.

Subsequent fields contain the terminal capabilities; any continued capability lines must be indented one tab from the left margin.

Although there is no defined order, it is suggested to write first boolean, then numeric, and then string capabilities, each sorted alphabetically without looking at lower or upper spelling. Capabilities of similar functions can be written in one line.

Example for:

```
Head line: vt|vt101|DEC VT 101 terminal in 80 character mode:\
Head line: Vt|vt101-w|DEC VT 101 terminal in (wide) 132 character mode:\
Boolean: :bs:\
Numeric: :co#80:\
String: :sr=\E[H:\
```

Boolean Capabilities

| | |
|----|--|
| 5i | Printer will not echo on screen |
| am | Automatic margins which means automatic line wrap |
| bs | Control-H (8 dec.) performs a backspace |
| bw | Backspace on left margin wraps to previous line and right margin |
| da | Display retained above screen |
| db | Display retained below screen |
| eo | A space erases all characters at cursor position |
| es | Escape sequences and special characters work in status line |
| gn | Generic device |
| hc | This is a hardcopy terminal |
| HC | The cursor is hard to see when not on bottom line |
| hs | Has a status line |
| hz | Hazeltine bug, the terminal can not print tilde characters |
| in | Terminal inserts null bytes, not spaces, to fill whitespace |
| km | Terminal has a meta key |
| mi | Cursor movement works in insert mode |
| ms | Cursor movement works in standout/underline mode |

| | |
|----|---|
| NP | No pad character |
| NR | ti does not reverse te |
| nx | No padding, must use XON/XOFF |
| os | Terminal can overstrike |
| ul | Terminal underlines although it can not overstrike |
| xb | Beehive glitch, f1 sends ESCAPE, f2 sends ^C |
| xn | Newline/wraparound glitch |
| xo | Terminal uses xon/xoff protocol |
| xs | Text typed over standout text will be displayed in standout |
| xt | Telera glitch, destructive tabs and odd standout mode |

Numeric Capabilities

| | |
|----|---|
| co | Number of columns |
| dB | Delay in milliseconds for backspace on hardcopy terminals |
| dC | Delay in milliseconds for carriage return on hardcopy terminals |
| dF | Delay in milliseconds for form feed on hardcopy terminals |
| dN | Delay in milliseconds for new line on hardcopy terminals |
| dT | Delay in milliseconds for tabulator stop on hardcopy terminals |
| dV | Delay in milliseconds for vertical tabulator stop on hardcopy terminals |
| it | Difference between tab positions |
| lh | Height of soft labels |
| lm | Lines of memory |
| lw | Width of soft labels |
| li | Number of lines |
| Nl | Number of soft labels |
| pb | Lowest baud rate which needs padding |
| sg | Standout glitch |
| ug | Underline glitch |
| vt | virtual terminal number |
| ws | Width of status line if different from screen width |

String Capabilities

| | |
|----|-------------------------|
| !1 | shifted save key |
| !2 | shifted suspend key |
| !3 | shifted undo key |
| #1 | shifted help key |
| #2 | shifted home key |
| #3 | shifted input key |
| #4 | shifted cursor left key |
| %0 | redo key |
| %1 | help key |
| %2 | mark key |
| %3 | message key |
| %4 | move key |
| %5 | next-object key |
| %6 | open key |
| %7 | options key |
| %8 | previous-object key |
| %9 | print key |
| %a | shifted message key |
| %b | shifted move key |
| %c | shifted next key |
| %d | shifted options key |
| %e | shifted previous key |
| %f | shifted print key |

| | |
|----|--|
| %g | shifted redo key |
| %h | shifted replace key |
| %i | shifted cursor right key |
| %j | shifted resume key |
| &0 | shifted cancel key |
| &1 | reference key |
| &2 | refresh key |
| &3 | replace key |
| &4 | restart key |
| &5 | resume key |
| &6 | save key |
| &7 | suspend key |
| &8 | undo key |
| &9 | shifted begin key |
| *0 | shifted find key |
| *1 | shifted command key |
| *2 | shifted copy key |
| *3 | shifted create key |
| *4 | shifted delete character |
| *5 | shifted delete line |
| *6 | select key |
| *7 | shifted end key |
| *8 | shifted clear line key |
| *9 | shifted exit key |
| @0 | find key |
| @1 | begin key |
| @2 | cancel key |
| @3 | close key |
| @4 | command key |
| @5 | copy key |
| @6 | create key |
| @7 | end key |
| @8 | enter/send key |
| @9 | exit key |
| al | Insert one line |
| AL | Insert %1 lines |
| ac | Pairs of block graphic characters to map alternate character set |
| ae | End alternative character set |
| as | Start alternative character set for block graphic characters |
| bc | Backspace, if not ^H |
| bl | Audio bell |
| bt | Move to previous tab stop |
| cb | Clear from beginning of line to cursor |
| cc | Dummy command character |
| cd | Clear to end of screen |
| ce | Clear to end of line |
| ch | Move cursor horizontally only to column %1 |
| cl | Clear screen and cursor home |
| cm | Cursor move to row %1 and column %2 (on screen) |
| CM | Move cursor to row %1 and column %2 (in memory) |
| cr | Carriage return |
| cs | Scroll region from line %1 to %2 |
| ct | Clear tabs |
| cv | Move cursor vertically only to line %1 |

| | |
|-----|--|
| dc | Delete one character |
| DC | Delete %1 characters |
| dl | Delete one line |
| DL | Delete %1 lines |
| dm | Begin delete mode |
| do | Cursor down one line |
| DO | Cursor down #1 lines |
| ds | Disable status line |
| eA | Enable alternate character set |
| ec | Erase %1 characters starting at cursor |
| ed | End delete mode |
| ei | End insert mode |
| ff | Formfeed character on hardcopy terminals |
| fs | Return character to its position before going to status line |
| F1 | The string sent by function key f11 |
| F2 | The string sent by function key f12 |
| F3 | The string sent by function key f13 |
| ... | ... |
| F9 | The string sent by function key f19 |
| FA | The string sent by function key f20 |
| FB | The string sent by function key f21 |
| ... | ... |
| FZ | The string sent by function key f45 |
| Fa | The string sent by function key f46 |
| Fb | The string sent by function key f47 |
| ... | ... |
| Fr | The string sent by function key f63 |
| hd | Move cursor a half line down |
| ho | Cursor home |
| hu | Move cursor a half line up |
| i1 | Initialization string 1 at login |
| i3 | Initialization string 3 at login |
| is | Initialization string 2 at login |
| ic | Insert one character |
| IC | Insert %1 characters |
| if | Initialization file |
| im | Begin insert mode |
| ip | Insert pad time and needed special characters after insert |
| iP | Initialization program |
| K1 | upper left key on keypad |
| K2 | center key on keypad |
| K3 | upper right key on keypad |
| K4 | bottom left key on keypad |
| K5 | bottom right key on keypad |
| k0 | Function key 0 |
| k1 | Function key 1 |
| k2 | Function key 2 |
| k3 | Function key 3 |
| k4 | Function key 4 |
| k5 | Function key 5 |
| k6 | Function key 6 |
| k7 | Function key 7 |
| k8 | Function key 8 |
| k9 | Function key 9 |

| | |
|-----|--|
| k; | Function key 10 |
| ka | Clear all tabs key |
| kA | Insert line key |
| kb | Backspace key |
| kB | Back tab stop |
| kC | Clear screen key |
| kd | Cursor down key |
| kD | Key for delete character under cursor |
| ke | turn keypad off |
| kE | Key for clear to end of line |
| kF | Key for scrolling forward/down |
| kh | Cursor home key |
| kH | Cursor hown down key |
| kI | Insert character/Insert mode key |
| kl | Cursor left key |
| kL | Key for delete line |
| kM | Key for exit insert mode |
| kN | Key for next page |
| kP | Key for previous page |
| kr | Cursor right key |
| kR | Key for scrolling backward/up |
| ks | Turn keypad on |
| kS | Clear to end of screen key |
| kt | Clear this tab key |
| kT | Set tab here key |
| ku | Cursor up key |
| l0 | Label of zeroth function key, if not f0 |
| l1 | Label of first function key, if not f1 |
| l2 | Label of first function key, if not f2 |
| ... | ... |
| la | Label of tenth function key, if not f10 |
| le | Cursor left one character |
| ll | Move cursor to lower left corner |
| LE | Cursor left %1 characters |
| LF | Turn soft labels off |
| LO | Turn soft labels on |
| mb | Start blinking |
| MC | Clear soft margins |
| md | Start bold mode |
| me | End all mode like so, us, mb, md and mr |
| mh | Start half bright mode |
| mk | Dark mode (Characters invisible) |
| ML | Set left soft margin |
| mm | Put terminal in meta mode |
| mo | Put terminal out of meta mode |
| mp | Turn on protected attribute |
| mr | Start reverse mode |
| MR | Set right soft margin |
| nd | Cursor right one character |
| nw | Carriage return command |
| pc | Padding character |
| pf | Turn printer off |
| pk | Program key %1 to send string %2 as if typed by user |
| pl | Program key %1 to execute string %2 in local mode |

| | |
|----|--|
| pn | Program soft label %1 to show string %2 |
| po | Turn the printer on |
| pO | Turn the printer on for %1 (<256) bytes |
| ps | Print screen contents on printer |
| px | Program key %1 to send string %2 to computer |
| r1 | Reset string 1 to set terminal to sane modes |
| r2 | Reset string 2 to set terminal to sane modes |
| r3 | Reset string 3 to set terminal to sane modes |
| RA | disable automatic margins |
| rc | Restore saved cursor position |
| rf | Reset string filename |
| RF | Request for input from terminal |
| RI | Cursor right %1 characters |
| rp | Repeat character %1 for %2 times |
| rP | Padding after character sent in replace mode |
| rs | Reset string |
| RX | Turn off XON/XOFF flow control |
| sa | Set %1 %2 %3 %4 %5 %6 %7 %8 %9 attributes |
| SA | enable automatic margins |
| sc | Save cursor position |
| se | End standout mode |
| sf | Normal scroll one line |
| SF | Normal scroll %1 lines |
| so | Start standout mode |
| sr | Reverse scroll |
| SR | scroll back %1 lines |
| st | Set tabulator stop in all rows at current column |
| SX | Turn on XON/XOFF flow control |
| ta | move to next hardware tab |
| tc | Read in terminal description from another entry |
| te | End program that uses cursor motion |
| ti | Begin program that uses cursor motion |
| ts | Move cursor to column %1 of status line |
| uc | Underline character under cursor and move cursor right |
| ue | End underlining |
| up | Cursor up one line |
| UP | Cursor up %1 lines |
| us | Start underlining |
| vb | Visible bell |
| ve | Normal cursor visible |
| vi | Cursor invisible |
| vs | Standout cursor |
| wi | Set window from line %1 to %2 and column %3 to %4 |
| XF | XOFF character if not ^S |

There are several ways of defining the control codes for string capabilities:

Every normal character represents itself, except '^', '\', and '%'.

A '^x' means Control-x. Control-A equals 1 decimal.

\x means a special code. x can be one of the following characters:

- E Escape (27)
- n Linefeed (10)
- r Carriage return (13)
- t Tabulation (9)
- b Backspace (8)

| | | |
|---|---|--|
| | f | Form feed (12) |
| | 0 | Null character. A \xxx specifies the octal character xxx. |
| i | | Increments parameters by one. |
| r | | Single parameter capability |
| + | | Add value of next character to this parameter and do binary output |
| 2 | | Do ASCII output of this parameter with a field with of 2 |
| d | | Do ASCII output of this parameter with a field with of 3 |
| % | | Print a '%' |

If you use binary output, then you should avoid the null character because it terminates the string. You should reset tabulator expansion if a tabulator can be the binary output of a parameter.

Warning:

The above metacharacters for parameters may be wrong, they document Minix termcap which may not be compatible with Linux termcap.

The block graphic characters can be specified by three string capabilities:

| | |
|----|---|
| as | start the alternative charset |
| ae | end it |
| ac | pairs of characters. The first character is the name of the block graphic symbol and the second characters is its definition. |

The following names are available:

| | |
|---|----------------------------|
| + | right arrow (>) |
| , | left arrow (<) |
| . | down arrow (v) |
| 0 | full square (#) |
| I | lantern (#) |
| - | upper arrow (^) |
| , | rhombus (+) |
| a | chess board (:) |
| f | degree (') |
| g | plus-minus (#) |
| h | square (#) |
| j | right bottom corner (+) |
| k | right upper corner (+) |
| l | left upper corner (+) |
| m | left bottom corner (+) |
| n | cross (+) |
| o | upper horizontal line (-) |
| q | middle horizontal line (-) |
| s | bottom horizontal line (_) |
| t | left tee (+) |
| u | right tee (+) |
| v | bottom tee (+) |
| w | normal tee (+) |
| x | vertical line () |
| ~ | paragraph (???) |

The values in parentheses are suggested defaults which are used by curses, if the capabilities are missing.

SEE ALSO

ncurses(3), **termcap(3)**, **terminfo(5)**

COLOPHON

This page is part of release 2.78 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at <http://www.kernel.org/doc/man-pages/>.

NAME

textblock – format stream to a justified text block

SYNOPSIS

edrc/lib/textblock [**-h**]

textblock [*blockwidth* [*hangingwidth*]]

AVAILABILITY

WA2L/edrc

DESCRIPTION

format a stream from **stdin** into a justified text block with an optional hanging first line.

OPTIONS

-h print usage message.

blockwidth width of the text block.

Default is **80**.

hangingwidth

width of the hanging indent.

Default is **0**.

EXIT STATUS

0 no error

4 usage displayed

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **indent(3)**

NOTES

-

BUGS

-

AUTHOR

textblock was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

textcolor – set the terminal text color

SYNOPSIS

edrc/lib/textcolor [**-h**]

textcolor *ATTRIBUTE FOREGROUND BACKGROUND*

AVAILABILITY

WA2L/edrc

DESCRIPTION

set the terminal text color.

OPTIONS

-h print usage message.

ATTRIBUTE

text attribute:

RESET, BRIGHT, DIM, UNDERLINE, BLINK, REVERSE or **HIDDEN**.

FOREGROUND

text color:

BLACK, RED, GREEN, YELLOW, BLUE, MAGENTA, CYAN or **WHITE**.

BACKGROUND

background color:

BLACK, RED, GREEN, YELLOW, BLUE, MAGENTA, CYAN or **WHITE**.

EXIT STATUS

0 no error

4 usage displayed

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **termcap(4)**, **tput_examples(3)**, **tty(1)**

NOTES

-

BUGS

-

AUTHOR

textcolor was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

thttpd - tiny/turbo/throttling HTTP server

SYNOPSIS

thttpd [-C *configfile*] [-p *port*] [-d *dir*] [-dd *data_dir*] [-r|-nor] [-s|-nos] [-v|-nov] [-g|-nog] [-u *user*] [-c *cgipat*] [-t *throttles*] [-h *host*] [-l *logfile*] [-i *pidfile*] [-T *charset*] [-P *P3P*] [-M *maxage*] [-V] [-D]

DESCRIPTION

thttpd is a simple, small, fast, and secure HTTP server. It doesn't have a lot of special features, but it suffices for most uses of the web, it's about as fast as the best full-featured servers (Apache, NCSA, Netscape), and it has one extremely useful feature (URL-traffic-based throttling) that no other server currently has.

OPTIONS

- C** Specifies a config-file to read. All options can be set either by command-line flags or in the config file. See below for details.
- p** Specifies an alternate port number to listen on. The default is 80. The config-file option name for this flag is "port", and the config.h option is DEFAULT_PORT.
- d** Specifies a directory to chdir() to at startup. This is merely a convenience - you could just as easily do a cd in the shell script that invokes the program. The config-file option name for this flag is "dir", and the config.h options are WEBDIR, USE_USER_DIR.
- r** Do a chroot() at initialization time, restricting file access to the program's current directory. If -r is the compiled-in default, then -nor disables it. See below for details. The config-file option names for this flag are "chroot" and "nochroot", and the config.h option is ALWAYS_CHROOT.
- dd** Specifies a directory to chdir() to after chrooting. If you're not chrooting, you might as well do a single chdir() with the -d flag. If you are chrooting, this lets you put the web files in a subdirectory of the chroot tree, instead of in the top level mixed in with the chroot files. The config-file option name for this flag is "data_dir".
- nos** Don't do explicit symbolic link checking. Normally, thttpd explicitly expands any symbolic links in filenames, to check that the resulting path stays within the original document tree. If you want to turn off this check and save some CPU time, you can use the -nos flag, however this is not recommended. Note, though, that if you are using the chroot option, the symlink checking is unnecessary and is turned off, so the safe way to save those CPU cycles is to use chroot. The config-file option names for this flag are "symlinkcheck" and "nosymlinkcheck".
- v** Do el-cheapo virtual hosting. If -v is the compiled-in default, then -nov disables it. See below for details. The config-file option names for this flag are "vhost" and "novhost", and the config.h option is ALWAYS_VHOST.
- g** Use a global passwd file. This means that every file in the entire document tree is protected by the single .htpasswd file at the top of the tree. Otherwise the semantics of the .htpasswd file are the same. If this option is set but there is no .htpasswd file in the top-level directory, then thttpd proceeds as if the option was not set - first looking for a local .htpasswd file, and if that doesn't exist either then serving the file without any password. If -g is the compiled-in default, then -nog disables it. The config-file option names for this flag are "globalpasswd" and "noglobalpasswd", and the config.h option is ALWAYS_GLOBAL_PASSWD.
- u** Specifies what user to switch to after initialization when started as root. The default is "nobody". The config-file option name for this flag is "user", and the config.h option is DEFAULT_USER.
- c** Specifies a wildcard pattern for CGI programs, for instance "***.cgi" or "/cgi-bin/*". See below for details. The config-file option name for this flag is "cgipat", and the config.h option is CGI_PATTERN.
- t** Specifies a file of throttle settings. See below for details. The config-file option name for this flag is "throttles".
- h** Specifies a hostname to bind to, for multihoming. The default is to bind to all hostnames supported on the local machine. See below for details. The config-file option name for this flag is

"host", and the config.h option is SERVER_NAME.

- l** Specifies a file for logging. If no -l argument is specified, thttpd logs via syslog(). If "-l /dev/null" is specified, thttpd doesn't log at all. The config-file option name for this flag is "logfile".
- i** Specifies a file to write the process-id to. If no file is specified, no process-id is written. You can use this file to send signals to thttpd. See below for details. The config-file option name for this flag is "pidfile".
- T** Specifies the character set to use with text MIME types. The default is iso-8859-1. The config-file option name for this flag is "charset", and the config.h option is DEFAULT_CHARSET.
- P** Specifies a P3P server privacy header to be returned with all responses. See <http://www.w3.org/P3P/> for details. Thttpd doesn't do anything at all with the string except put it in the P3P: response header. The config-file option name for this flag is "p3p".
- M** Specifies the number of seconds to be used in a "Cache-Control: max-age" header to be returned with all responses. An equivalent "Expires" header is also generated. The default is no Cache-Control or Expires headers, which is just fine for most sites. The config-file option name for this flag is "max_age".
- V** Shows the current version info.
- D** This was originally just a debugging flag, however it's worth mentioning because one of the things it does is prevent thttpd from making itself a background daemon. Instead it runs in the foreground like a regular program. This is necessary when you want to run thttpd wrapped in a little shell script that restarts it if it exits.

CONFIG-FILE

All the command-line options can also be set in a config file. One advantage of using a config file is that the file can be changed, and thttpd will pick up the changes with a restart.

The syntax of the config file is simple, a series of "option" or "option=value" separated by whitespace. The option names are listed above with their corresponding command-line flags.

CHROOT

chroot() is a system call that restricts the program's view of the filesystem to the current directory and directories below it. It becomes impossible for remote users to access any file outside of the initial directory. The restriction is inherited by child processes, so CGI programs get it too. This is a very strong security measure, and is recommended. The only downside is that only root can call chroot(), so this means the program must be started as root. However, the last thing it does during initialization is to give up root access by becoming another user, so this is safe.

The program can also be compile-time configured to always do a chroot(), without needing the -r flag.

Note that with some other web servers, such as NCSA httpd, setting up a directory tree for use with chroot() is complicated, involving creating a bunch of special directories and copying in various files. With thttpd it's a lot easier, all you have to do is make sure any shells, utilities, and config files used by your CGI programs and scripts are available. If you have CGI disabled, or if you make a policy that all CGI programs must be written in a compiled language such as C and statically linked, then you probably don't have to do any setup at all.

However, one thing you should do is tell syslogd about the chroot tree, so that thttpd can still generate syslog messages. Check your system's syslogd man page for how to do this. In FreeBSD you would put something like this in /etc/rc.conf:

```
syslogd_flags="-l /usr/local/www/data/dev/log"
```

Substitute in your own chroot tree's pathname, of course. Don't worry about creating the log socket, syslogd wants to do that itself. (You may need to create the dev directory.) In Linux the flag is -a instead of -l, and there may be other differences.

Relevant config.h option: ALWAYS_CHROOT.

CGI

thttpd supports the CGI 1.1 spec.

In order for a CGI program to be run, its name must match the pattern specified either at compile time or on the command line with the `-c` flag. This is a simple shell-style filename pattern. You can use `*` to match any string not including a slash, or `**` to match any string including slashes, or `?` to match any single character. You can also use multiple such patterns separated by `|`. The patterns get checked against the filename part of the incoming URL. Don't forget to quote any wildcard characters so that the shell doesn't mess with them.

Restricting CGI programs to a single directory lets the site administrator review them for security holes, and is strongly recommended. If there are individual users that you trust, you can enable their directories too.

If no CGI pattern is specified, neither here nor at compile time, then CGI programs cannot be run at all. If you want to disable CGI as a security measure, that's how you do it, just comment out the patterns in the config file and don't run with the `-c` flag.

Note: the current working directory when a CGI program gets run is the directory that the CGI program lives in. This isn't in the CGI 1.1 spec, but it's what most other HTTP servers do.

Relevant config.h options: `CGI_PATTERN`, `CGI_TIMELIMIT`, `CGI_NICE`, `CGI_PATH`, `CGI_LD_LIBRARY_PATH`, `CGIBINDIR`.

BASIC AUTHENTICATION

Basic Authentication is available as an option at compile time. If enabled, it uses a password file in the directory to be protected, called `.htpasswd` by default. This file is formatted as the familiar colon-separated username/encrypted-password pair, records delimited by newlines. The protection does not carry over to subdirectories. The utility program `htpasswd(1)` is included to help create and modify `.htpasswd` files.

Relevant config.h option: `AUTH_FILE`

THROTTLING

The throttle file lets you set maximum byte rates on URLs or URL groups. You can optionally set a minimum rate too. The format of the throttle file is very simple. A `#` starts a comment, and the rest of the line is ignored. Blank lines are ignored. The rest of the lines should consist of a pattern, whitespace, and a number. The pattern is a simple shell-style filename pattern, using `?/**/*`, or multiple such patterns separated by `|`.

The numbers in the file are byte rates, specified in units of bytes per second. For comparison, a v.90 modem gives about 5000 B/s depending on compression, a double-B-channel ISDN line about 12800 B/s, and a T1 line is about 150000 B/s. If you want to set a minimum rate as well, use number-number.

Example:

```
# throttle file for www.acme.com

**          2000-100000 # limit total web usage to 2/3 of our T1,
                # but never go below 2000 B/s
**.jpg|.gif 50000 # limit images to 1/3 of our T1
**.mpg      20000 # and movies to even less
jef/**      20000 # jef's pages are too popular
```

Throttling is implemented by checking each incoming URL filename against all of the patterns in the throttle file. The server accumulates statistics on how much bandwidth each pattern has accounted for recently (via a rolling average). If a URL matches a pattern that has been exceeding its specified limit, then the data returned is actually slowed down, with pauses between each block. If that's not possible (e.g. for CGI programs) or if the bandwidth has gotten way larger than the limit, then the server returns a special code saying 'try again later'.

The minimum rates are implemented similarly. If too many people are trying to fetch something at the same time, throttling may slow down each connection so much that it's not really useable. Furthermore, all those slow connections clog up the server, using up file handles and connection slots. Setting a minimum

rate says that past a certain point you should not even bother - the server returns the "try again later" code and the connection isn't even started.

There is no provision for setting a maximum connections/second throttle, because throttling a request uses as much CPU as handling it, so there would be no point. There is also no provision for throttling the number of simultaneous connections on a per-URL basis. However you can control the overall number of connections for the whole server very simply, by setting the operating system's per-process file descriptor limit before starting thttpd. Be sure to set the hard limit, not the soft limit.

MULTIHOSTING

Multihoming means using one machine to serve multiple hostnames. For instance, if you're an internet provider and you want to let all of your customers have customized web addresses, you might have `www.joe.acme.com`, `www.jane.acme.com`, and your own `www.acme.com`, all running on the same physical hardware. This feature is also known as "virtual hosts". There are three steps to setting this up.

One, make DNS entries for all of the hostnames. The current way to do this, allowed by HTTP/1.1, is to use CNAME aliases, like so:

```
www.acme.com IN A 192.100.66.1
www.joe.acme.com IN CNAME www.acme.com
www.jane.acme.com IN CNAME www.acme.com
```

However, this is incompatible with older HTTP/1.0 browsers. If you want to stay compatible, there's a different way - use A records instead, each with a different IP address, like so:

```
www.acme.com IN A 192.100.66.1
www.joe.acme.com IN A 192.100.66.200
www.jane.acme.com IN A 192.100.66.201
```

This is bad because it uses extra IP addresses, a somewhat scarce resource. But if you want people with older browsers to be able to visit your sites, you still have to do it this way.

Step two. If you're using the modern CNAME method of multihoming, then you can skip this step. Otherwise, using the older multiple-IP-address method you must set up IP aliases or multiple interfaces for the extra addresses. You can use `ifconfig(8)`'s `alias` command to tell the machine to answer to all of the different IP addresses. Example:

```
ifconfig le0 www.acme.com
ifconfig le0 www.joe.acme.com alias
ifconfig le0 www.jane.acme.com alias
```

If your OS's version of `ifconfig` doesn't have an `alias` command, you're probably out of luck (but see <http://www.acme.com/software/thttpd/notes.html>).

Third and last, you must set up thttpd to handle the multiple hosts. The easiest way is with the `-v` flag, or the `ALWAYS_VHOST` config.h option. This works with either CNAME multihosting or multiple-IP multihosting. What it does is send each incoming request to a subdirectory based on the hostname it's intended for. All you have to do in order to set things up is to create those subdirectories in the directory where thttpd will run. With the example above, you'd do like so:

```
mkdir www.acme.com www.joe.acme.com www.jane.acme.com
```

If you're using old-style multiple-IP multihosting, you should also create symbolic links from the numeric addresses to the names, like so:

```
ln -s www.acme.com 192.100.66.1
ln -s www.joe.acme.com 192.100.66.200
ln -s www.jane.acme.com 192.100.66.201
```

This lets the older HTTP/1.0 browsers find the right subdirectory.

There's an optional alternate step three if you're using multiple-IP multihosting: run a separate thttpd process for each hostname, using the `-h` flag to specify which one is which. This gives you more flexibility, since you can run each of these processes in separate directories, with different throttle files, etc. Example:

```
thttpd -r -d /usr/www -h www.acme.com
thttpd -r -d /usr/www/joe -u joe -h www.joe.acme.com
thttpd -r -d /usr/www/jane -u jane -h www.jane.acme.com
```

But remember, this multiple-process method does not work with CNAME multihosting - for that, you must

use a single thttpd process with the -v flag.

CUSTOM ERRORS

thttpd lets you define your own custom error pages for the various HTTP errors. There's a separate file for each error number, all stored in one special directory. The directory name is "errors", at the top of the web directory tree. The error files should be named "errNNN.html", where NNN is the error number. So for example, to make a custom error page for the authentication failure error, which is number 401, you would put your HTML into the file "errors/err401.html". If no custom error file is found for a given error number, then the usual built-in error page is generated.

If you're using the virtual hosts option, you can also have different custom error pages for each different virtual host. In this case you put another "errors" directory in the top of that virtual host's web tree. thttpd will look first in the virtual host errors directory, and then in the server-wide errors directory, and if neither of those has an appropriate error file then it will generate the built-in error.

NON-LOCAL REFERERS

Sometimes another site on the net will embed your image files in their HTML files, which basically means they're stealing your bandwidth. You can prevent them from doing this by using non-local referer filtering. With this option, certain files can only be fetched via a local referer. The files have to be referenced by a local web page. If a web page on some other site references the files, that fetch will be blocked. There are three config-file variables for this feature:

urlpat A wildcard pattern for the URLs that should require a local referer. This is typically just image files, sound files, and so on. For example:

```
urlpat=*.jpg|*.gif|*.au|*.wav
```

For most sites, that one setting is all you need to enable referer filtering.

noemptyreferers

By default, requests with no referer at all, or a null referer, or a referer with no apparent hostname, are allowed. With this variable set, such requests are disallowed.

localpat

A wildcard pattern that specifies the local host or hosts. This is used to determine if the host in the referer is local or not. If not specified it defaults to the actual local hostname.

SYMLINKS

thttpd is very picky about symbolic links. Before delivering any file, it first checks each element in the path to see if it's a symbolic link, and expands them all out to get the final actual filename. Along the way it checks for things like links with ".." that go above the server's directory, and absolute symlinks (ones that start with a /). These are prohibited as security holes, so the server returns an error page for them. This means you can't set up your web directory with a bunch of symlinks pointing to individual users' home web directories. Instead you do it the other way around - the user web directories are real subdirs of the main web directory, and in each user's home dir there's a symlink pointing to their actual web dir.

The CGI pattern is also affected - it gets matched against the fully-expanded filename. So, if you have a single CGI directory but then put a symbolic link in it pointing somewhere else, that won't work. The CGI program will be treated as a regular file and returned to the client, instead of getting run. This could be confusing.

PERMISSIONS

thttpd is also picky about file permissions. It wants data files (HTML, images) to be world readable. Readable by the group that the thttpd process runs as is not enough - thttpd checks explicitly for the world-readable bit. This is so that no one ever gets surprised by a file that's not set world-readable and yet somehow is readable by the HTTP server and therefore the *whole* world.

The same logic applies to directories. As with the standard Unix "ls" program, thttpd will only let you look at the contents of a directory if its read bit is on; but as with data files, this must be the world-read bit, not just the group-read bit.

thttpd also wants the execute bit to be *off* for data files. A file that is marked executable but doesn't match the CGI pattern might be a script or program that got accidentally left in the wrong directory.

Allowing people to fetch the contents of the file might be a security breach, so this is prohibited. Of course if an executable file **does** match the CGI pattern, then it just gets run as a CGI.

In summary, data files should be mode 644 (rw-r--r--), directories should be 755 (rwxr-xr-x) if you want to allow indexing and 711 (rwx--x--x) to disallow it, and CGI programs should be mode 755 (rwxr-xr-x) or 711 (rwx--x--x).

LOGS

thttpd does all of its logging via syslog(3). The facility it uses is configurable. Aside from error messages, there are only a few log entry types of interest, all fairly similar to CERN Common Log Format:

```
Aug 6 15:40:34 acme thttpd[583]: 165.113.207.103 - - "GET /file" 200 357
```

```
Aug 6 15:40:43 acme thttpd[583]: 165.113.207.103 - - "HEAD /file" 200 0
```

```
Aug 6 15:41:16 acme thttpd[583]: referer http://www.acme.com/ -> /dir
```

```
Aug 6 15:41:16 acme thttpd[583]: user-agent Mozilla/1.1N
```

The package includes a script for translating these log entries into CERN-compatible files. Note that thttpd does not translate numeric IP addresses into domain names. This is both to save time and as a minor security measure (the numeric address is harder to spoof).

Relevant config.h option: LOG_FACILITY.

If you'd rather log directly to a file, you can use the `-l` command-line flag. But note that error messages still go to syslog.

SIGNALS

thttpd handles a couple of signals, which you can send via the standard Unix kill(1) command:

INT,TERM

These signals tell thttpd to shut down immediately. Any requests in progress get aborted.

USR1 This signal tells thttpd to shut down as soon as it's done servicing all current requests. In addition, the network socket it uses to accept new connections gets closed immediately, which means a fresh thttpd can be started up immediately.

USR2 This signal tells thttpd to generate the statistics syslog messages immediately, instead of waiting for the regular hourly update.

HUP This signal tells thttpd to close and re-open its (non-syslog) log file, for instance if you rotated the logs and want it to start using the new one. This is a little tricky to set up correctly, for instance if you are using chroot() then the log file must be within the chroot tree, but it's definitely doable.

SEE ALSO

redirect(8), ssi(8), makeweb(1), htpasswd(1), syslogtocern(8), weblog_parse(1), http_get(1)

THANKS

Many thanks to contributors, reviewers, testers: John LoVerso, Jordan Hayes, Chris Torek, Jim Thompson, Barton Schaffer, Geoff Adams, Dan Kegel, John Hascall, Bennett Todd, KIKUCHI Takahiro, Catalin Ionescu. Special thanks to Craig Leres for substantial debugging and development, and for not complaining about my coding style very much.

AUTHOR

Copyright © 1995,1998,1999,2000 by Jef Poskanzer <jef@mail.acme.com>. All rights reserved.

NAME

timer – print seconds since the Epoch

SYNOPSIS

Version = 1.0.00:

edrc/bin/timer [**-h**]

timer [[**-d**] *seconds*]

Version >= 1.0.01:

edrc/bin/timer [**-h** | **-V**]

timer [[**-d**] *seconds* | *YYYY-MM-DD* | "*YYYY-MM-DD hh:mm:ss*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the number of seconds since the Epoch or return the date based on the given number of seconds since the Epoch.

Since version 1.0.01 of the **timer** command, the Epoch for a given date or date and time can be resolved.

OPTIONS

-d return the date and time in the standard format as printed when the **date**(1) command is issued without options.

seconds seconds to return the date for it. If the **-d** option is not specified, the date is returned in military format (**YYYY-MM-DD hh:mm:ss**).

YYYY-MM-DD
return the Epoch for the given date at *00:00:00*.

"YYYY-MM-DD hh:mm:ss"
return the Epoch for the given date and time.

ENVIRONMENT

\$TZ time zone.

EXIT STATUS

| | |
|-----------|---|
| 4 | usage printed |
| 11 | no error, seconds since Epoch printed. |
| 16 | no error, seconds of given date time printed. |
| 20 | no error, date based on given seconds printed. |
| 25 | no error, date based on given seconds printed in standard format. |

FILES

-

EXAMPLES

The following examples are given as if they are all executed at the same point in time.

1) print the date

```
[ / ]
[ root@acme007 ][*edrc*/bash]: date

Sun Sep 29 22:36:04 CEST 2008
```

2) print seconds since the Epoch

```
[ / ]
[ root@acme007 ][*edrc*/bash]: timer

1222720564
```

3) calculate the timer of yesterday

```
[ / ]
[ root@acme007 ][*edrc*/bash]: expr `timer` - 60 60 24

1222634164
```

4) print the date based on a (calculated) timer

```
[ / ]
[ root@acme007 ][*edrc*/bash]: timer -d 1222634164

Sun Sep 28 22:36:04 2008
```

5) print the date in military format based on a (calculated) timer

```
[ / ]  
[ root@acme007 ][*edrc*/bash]: timer 1222634164  
  
2008-09-28 22:36:04
```

SEE ALSO

edrcintro(1), date(1), seconds(3), days(3)

NOTES

-

BUGS

-

AUTHOR

timer was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2023 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

timezone – return the timezone

SYNOPSIS

edrc/lib/timezone [**-h** | **-v**]

timezone

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the timezone as defined system wide, set in the **\$TZ** environment variable or set in the configuration file **timezone.cfg**.

This command is used to return the time zone using a cross operating system compatible interface, due to the fact that on many unix flavors the setting of the time zone is implemented differently.

The following pseudo code explains the precise resolution of the return value of **timezone** (**TIMEZONE** and **TIMEZONE_DEFAULT** are settings in the **timezone.cfg** file):

```
if TIMEZONE <> "" then
    return TIMEZONE
else
    if $TZ <> "" then
        return $TZ
    else
        if 'system wide time zone setting' <> "" then
            return 'system wide time zone'
        else
            if TIMEZONE_DEFAULT <> "" then
                return TIMEZONE_DEFAULT
            else
                return "GMT"
            endif
        endif
    endif
endif
```

See also **timezone.cfg**(4) for the description of the possible settings and the influence to the **timezone** return value.

OPTIONS

- h** print usage message.
- v** print verbose information.

ENVIRONMENT

- \$TZ** Timezone environment variable setting.

EXIT STATUS

- 0** no error.
- 4** usage message printed.
- 6** config file does not exist.

FILES**/etc/TIMEZONE**

system wide time zone setting on Solaris. This file must contain the entry **TZ=a_timezone** .

/etc/default/tz

time zone setting on HPUX. This file must contain a single line specifying the time zone entry *a_timezone*.

/etc/localtime

On Linux **/etc/localtime** defines the time zone. Normally this file is a symbolic link to the real time zone file in **/usr/share/zoneinfo**.

/etc/default/tz

time zone setting on all other operating systems. This file must contain a single line specifying the time zone entry *a_timezone*.

edrc/etc/timezone.cfg

config file for **timezone**.

edrc/etc/timezone.dat

timezone map file **timezone** on Linux.

EXAMPLES

-

SEE ALSO

edrcintro(1), **timezone.cfg(4)**, **timezone.dat(4)**, **TIMEZONE(4)**

NOTES

The **timezone** command is used internally in many WA2L/edrc commands to ensure a consistent output of all dates within the whole package.

BUGS

-

AUTHOR

timezone was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

timezone.cfg – configuration file for timezone

SYNOPSIS

edrc/etc/timezone.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **timezone** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**TIMEZONE**

Time zone. This value is returned when the **timezone** command is invoked.

If this setting is not set, the content of the **\$TZ** environment variable is used, if the **\$TZ** environment variable is also not set, the system wide time zone setting is used. If the system wide time zone is not set, the setting in **TIMEZONE_DEFAULT** is used.

Example: **TIMEZONE=""**

Default: **TIMEZONE=TIMEZONE_DEFAULT**

TIMEZONE_DEFAULT

Default time zone if **TIMEZONE** is not set, the **\$TZ** environment variable is not set and the system wide time zone could not be resolved.

If this setting is also not set, *GMT* is used as a final default.

Example: **TIMEZONE_DEFAULT="MET-1METDST"**

Default: TIMEZONE_DEFAULT=GMT

SEE ALSO

edrcintro(1), **timezone**(3)

NOTES

-

BUGS

-

AUTHOR

timezone.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

timezone.dat – timezone configuration file for timezone

SYNOPSIS

edrc/etc/timezone.dat

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the timezone configuration definition for the **timezone(3)** command.

FILEFORMAT

The fileformat is a list of definitions that have the format

OSID:TIMEZONE:ALIAS:

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

OSID operating system id as returned by **osid(3)**.

TIMEZONE

timezone string as used in the **\$TZ** environment variable.

ALIAS timezone alias.

EXAMPLES

```
#
# timezone.dat - timezone mapping configuration file
#
# [00] 11.05.2010 CWa   Initial Version
# [01] 24.09.2010 CWa   +Linux-ia
#
#
# Fileformat:
```

```
#
# OSID:TIMEZONE:ALIAS:
#
Linux:ACST:America/Eirunepe:
Linux:ACST:America/Porto_Acre:
Linux:ACST:America/Rio_Branco:
Linux:ACST:Brazil/Acre:
Linux:ACST:posix/America/Eirunepe:
Linux:ACST:posix/America/Porto_Acre:
Linux:ACST:posix/America/Rio_Branco:
Linux:ACST:posix/Brazil/Acre:
Linux:ACST:right/America/Eirunepe:
Linux:ACST:right/America/Porto_Acre:
Linux:ACST:right/America/Rio_Branco:
Linux:ACST:right/Brazil/Acre:
Linux:ACT:America/Eirunepe:
Linux:ACT:America/Porto_Acre:
Linux:ACT:America/Rio_Branco:
Linux:ACT:Brazil/Acre:
Linux:ACT:posix/America/Eirunepe:
Linux:ACT:posix/America/Porto_Acre:
```

SEE ALSO

edrcintro(1), **timezone**(4), **tzdump**(3)

NOTES

-

BUGS

-

AUTHOR

timezone.dat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

title – set terminal window title

SYNOPSIS

edrc/bin/title [**-h**]

title [**short** | **long** | "*text*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

set the title of the terminal window. There are two predefined titles that can be chosen using the **short** or **long** option. It is also possible to define an own title text.

The title of the terminal window is only set, if you are logged on to a tty using a **/dev/pts/*** device.

OPTIONS

-h usage message.

short set title to 'user@hostname'. This is also the default if no option is specified.
Example: 'root@acme001'

long set title to 'user@hostname -- Customer: [ENVIRONMENT] Description' where Customer, ENVIRONMENT and Description is resolved by **server_environment(3)** internally.
Example: 'root@acme001 -- ACME: [PRODUCTION] Beringen, Acme Production'

"text" free form title text.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO**edrcintro(1)**, **hostname(1)**, **server_environment(3)****NOTES**

The **title** command is also used within **edrc** to set the terminal window title.

BUGS

-

AUTHOR

title was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

today – return the date of today

SYNOPSIS

edrc/lib/today [**-h**]

today ["*date_format*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the date of today. By default the date is returned in the format *YYYY-MM-DD*.

OPTIONS

-h print usage message.

"*date_format*"

date format as known from the **date**(1) command.

The following format place holders are the most compatible ones:

%Y year (1970...)

%y last two digits of year (00..99)

%m month (01..12)

%d day of month (01..31)

%H hour (00..23)

%M minute (00..59)

%S second (00..60)

%w day of week (0..6); 0 represents Sunday

ENVIRONMENT

\$TZ Timezone setting.

EXIT STATUS

0 no error.

today(3)

Library Commands

today(3)

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **date(1)**, **days(3)**, **is_weekend(3)**, **seconds(3)**, **timer(1)**, **tomorrow(3)**, **yesterday(3)**

NOTES

-

BUGS

-

AUTHOR

today was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tolower – return given string in lower case

SYNOPSIS

edrc/lib/tolower [**-h**]

tolower ["*string*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the string specified in *string* in lower case to **stdout**.

OPTIONS

-h usage message.

"*string*" string to be translated to lower case.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **toupper(3)**

NOTES

-

BUGS

-

AUTHOR

tolower was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tomorrow – return the date of tomorrow

SYNOPSIS

edrc/lib/tomorrow [**-h**]

tomorrow ["*date_format*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the date of tomorrow. By default the date is returned in the format *YYYY-MM-DD*.

OPTIONS

-h print usage message.

"*date_format*"

date format as known from the **date**(1) command.

The following format place holders are the most compatible ones:

%Y year (1970...)

%y last two digits of year (00..99)

%m month (01..12)

%d day of month (01..31)

%H hour (00..23)

%M minute (00..59)

%S second (00..60)

%w day of week (0..6); 0 represents Sunday

ENVIRONMENT

\$TZ Timezone setting.

EXIT STATUS

0 no error.

2 date could not be resolved.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **date(1)**, **days(3)**, **seconds(3)**, **timer(1)**, **today(3)**, **yesterday(3)**

NOTES

-

BUGS

-

AUTHOR

tomorrow was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tpl – select and print template file

SYNOPSIS

edrc/bin/tpl [*select*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

give a selection of template files to be printed to **stdout**.

The **tpl** command can also be called directly from within **vi** when editing a recovery script, as:

```
1 #!/bin/bash
2 #
3 # new:new_script - <enter the description here>
4 #
5 # [00] 22.10.2021 root      Initial Version
6 #
7 #
8 test "$DEBUG" = True && set -x
9
~
~
~
:.. !tpl
```

After receiving the template data in **vi**, press **[Ctrl] + [L]** to redraw the terminal.

OPTIONS

-h usage.

select pre-select files from list of template files.

ENVIRONMENT

\$TPL_CFG

additional configuration file (to **\$EDRC_SCRIPTS_BASEDIR/tpl.cfg** and **edrc/etc/tpl.cfg**) with template locations for **tpl**.

EXIT STATUS

- 0** no error.
- 4** usage printed.

FILES**edrc/etc/tpl.cfg**

list of files to be listed in the selection. The files are relative to the application root.

edrc/var/samples/templates/

some default templates distributed with WA2L/edrc.

\$EDRC_SCRIPTS_BASEDIR/tpl.cfg

list of files to be listed in the selection. The files are relative to the recovery scripts base directory.

\$TPL_CFG

additional list of files to be listed in the selection. The files are relative to the configuration file defined in **\$TPL_CFG**.

EXAMPLES

-

SEE ALSO

edrcintro(1), **logtail(1)**, **logview(1)**, **outex(1)**, **tpl.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

tpl was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tpl.cfg – template file list for tpl

SYNOPSIS

edrc/etc/tpl.cfg

\$EDRC_SCRIPTS_BASEDIR/tpl.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the template file definition for the **tpl** command.

FILEFORMAT

The fileformat is a plain list of template files, where wildcards as understood by the **ls(1)** command are supported.

The template files must not exist on the system, the **tpl** command resolves the existing template files and only presents the existing readable ones to the user.

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

The template files list are relative to the root of the WA2L/edrc package installation, as returned by the **app-root(3)** or the **\$EDRC_SCRIPTS_BASEDIR** depending on the location of the **tpl.cfg** file.

OPTIONS

-

EXAMPLES

```
#
# tpl.cfg - list of template files for tpl
#
# [00] 22.10.2021 CWa Initial Version
#
var/samples/templates/*
```

SEE ALSO

edrcintro(1), **tpl(1)**

NOTES

-

BUGS

-

AUTHOR

tpl.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

top – display and update information about the top cpu processes

SYNOPSIS

top [**-CISTbcinqv**] [**-dcount**] [**-stime**] [**-o field**] [**-Uusername**] [*number*]

DESCRIPTION

Top displays the top 10 processes on the system and periodically updates this information. Raw cpu percentage is used to rank the processes. If *number* is given, then the top *number* processes will be displayed instead of the default.

Top makes a distinction between terminals that support advanced capabilities and those that do not. This distinction affects the choice of defaults for certain options. In the remainder of this document, an “intelligent” terminal is one that supports cursor addressing, clear screen, and clear to end of line. Conversely, a “dumb” terminal is one that does not support such features. If the output of *top* is redirected to a file, it acts as if it were being run on a dumb terminal.

OPTIONS

- C** Turn off the use of color in the display.
- I** Do not display idle processes. By default, *top* displays both active and idle processes.
- S** Show system processes in the display. Normally, system processes such as the pager and the swapper are not shown. This option makes them visible.
- T** List all available color tags and the current set of tests used for color highlighting, then exit.
- b** Use “batch” mode. In this mode, all input from the terminal is ignored. Interrupt characters (such as ^C and ^_) still have an effect. This is the default on a dumb terminal, or when the output is not a terminal.
- c** Show the full command line for each process. Default is to show just the command name. This option is not supported on all platforms.
- i** Use “interactive” mode. In this mode, any input is immediately read for processing. See the section on “Interactive Mode” for an explanation of which keys perform what functions. After the command is processed, the screen will immediately be updated, even if the command was not understood. This mode is the default when standard output is an intelligent terminal.
- n** Use “non-interactive” mode. This is identical to “batch” mode.
- q** Renice *top* to -20 so that it will run faster. This can be used when the system is being very sluggish to improve the possibility of discovering the problem. This option can only be used by root.
- u** Do not take the time to map uid numbers to usernames. Normally, *top* will read as much of the file “/etc/passwd” as is necessary to map all the user id numbers it encounters into login names. This option disables all that, while possibly decreasing execution time. The uid numbers are displayed instead of the names.
- v** Write version number information to stderr then exit immediately. No other processing takes place when this option is used. To see current revision information while *top* is running, use the help command “?”.
- dcount** Show only *count* displays, then exit. A display is considered to be one update of the screen. This option allows the user to select the number of displays he wants to see before *top* automatically exits. For intelligent terminals, no upper limit is set. The default is 1 for dumb terminals.
- stime** Set the delay between screen updates to *time* seconds. The default delay between updates is 5 seconds.
- o field** Sort the process display area on the specified field. The field name is the name of the column as seen in the output, but in lower case. Likely values are “cpu”, “size”, “res”, and “time”, but may vary on different operating systems. Note that not all operating systems support this option.

-Uusername

Show only those processes owned by *username*. This option currently only accepts usernames and will not understand uid numbers.

Both *count* and *number* fields can be specified as “infinite”, indicating that they can stretch as far as possible. This is accomplished by using any proper prefix of the keywords “infinity”, “maximum”, or “all”. The default for *count* on an intelligent terminal is, in fact, **infinity**.

The environment variable **TOP** is examined for options before the command line is scanned. This enables a user to set his or her own defaults. The number of processes to display can also be specified in the environment variable **TOP**. The options **-C**, **-I**, **-S**, and **-u** are actually toggles. A second specification of any of these options will negate the first. Thus a user who has the environment variable **TOP** set to “**-I**” may use the command “top **-I**” to see idle processes.

INTERACTIVE MODE

When *top* is running in “interactive mode”, it reads commands from the terminal and acts upon them accordingly. In this mode, the terminal is put in “CBREAK”, so that a character will be processed as soon as it is typed. Almost always, a key will be pressed when *top* is between displays; that is, while it is waiting for *time* seconds to elapse. If this is the case, the command will be processed and the display will be updated immediately thereafter (reflecting any changes that the command may have specified). This happens even if the command was incorrect. If a key is pressed while *top* is in the middle of updating the display, it will finish the update and then process the command. Some commands require additional information, and the user will be prompted accordingly. While typing this information in, the user’s erase and kill keys (as set up by the command *stty*) are recognized, and a newline terminates the input.

These commands are currently recognized (^L refers to control-L):

- ^L** Redraw the screen.
- h** or **?** Display a summary of the commands (help screen). Version information is included in this display.
- q** Quit *top*.
- d** Change the number of displays to show (prompt for new number). Remember that the next display counts as one, so typing **d1** will make *top* show one final display and then immediately exit.
- n** or **#** Change the number of processes to display (prompt for new number).
- s** Change the number of seconds to delay between displays (prompt for new number).
- k** Send a signal (“kill” by default) to a list of processes. This acts similarly to the command *kill(1)*.
- r** Change the priority (the “nice”) of a list of processes. This acts similarly to the command *renice(8)*.
- u** Display only processes owned by a specific username (prompt for username). If the username specified is simply “+”, then processes belonging to all users will be displayed.
- o** Change the order in which the display is sorted. This command is not available on all systems. The sort key names vary from system to system but usually include: “cpu”, “res”, “size”, “time”. The default is cpu.
- e** Display a list of system errors (if any) generated by the last **kill** or **renice** command.
- i** (or **I**) Toggle the display of idle processes.
- c** Toggle the display of the full command line.
- C** Toggle the use of color in the display.

THE DISPLAY

The actual display varies depending on the specific variant of Unix that the machine is running. This description may not exactly match what is seen by *top* running on this particular machine. Differences are listed at the end of this manual entry.

The top few lines of the display show general information about the state of the system, including the last process id assigned to a process (on most systems), the three load averages, the current time, the number of existing processes, the number of processes in each state (sleeping, running, starting, zombies, and stopped), and a percentage of time spent in each of the processor states (user, nice, system, and idle). It also includes information about physical and virtual memory allocation.

The remainder of the screen displays information about individual processes. This display is similar in spirit to *ps(1)* but it is not exactly the same. The columns displayed by top will differ slightly between operating systems. Generally, the following fields are displayed:

PID The process id.

USERNAME

Username of the process's owner (if **-u** is specified, a UID column will be substituted for USERNAME).

PRI Current priority of the process.

NICE Nice amount in the range -20 to 20, as established by the use of the command *nice*.

SIZE Total size of the process (text, data, and stack) given in kilobytes.

RES Resident memory: current amount of process memory that resides in physical memory, given in kilobytes.

STATE Current state (typically one of "sleep", "run", "idl", "zomb", or "stop").

TIME Number of system and user cpu seconds that the process has used.

CPU Percentage of available cpu time used by this process.

COMMAND

Name of the command that the process is currently running.

COLOR

Top supports the use of ANSI color in its output. By default, color is available but not used. The environment variable **TOPCOLORS** specifies colors to use and conditions for which they should be used. At the present time, only numbers in the summary display area can be colored. In a future version it will be possible to highlight numbers in the process display area as well. The environment variable is the only way to specify color: there is no equivalent command line option. Note that the environment variable **TOP-COLOURS** is also understood. The British spelling takes precedence. The use of color only works on terminals that understand and process ANSI color escape sequences.

The environment variable is a sequence of color specifications, separated by colons. Each specification takes the form *tag*=*min*,*max*#*code* where *tag* is the name of the value to check, *min* and *max* specify a range for the value, and *code* is an ANSI color code. Multiple color codes can be listed and separated with semi-colons. A missing *min* implies the lowest possible value (usually 0) and a missing *max* implies infinity. The comma must always be present. When specifying numbers for load averages, they should be multiplied by 100. For example, the specification **1min=500,1000#31** indicates that a 1 minute load average between 5 and 10 should be displayed in red. Color attributes can be combined. For example, the specification **5min=1000,#37;41** indicates that a 5 minute load average higher than 10 should be displayed with white characters on a red background. A special tag named *header* is used to control the color of the header for process display. It should be specified with no lower and upper limits, specifically **header=,#** followed by the ANSI color code.

You can see a list of color codes recognized by this installation of top with the **-T** option. This will also show the current set of tests used for color highlighting, as specified in the environment.

AUTHOR

William LeFebvre

ENVIRONMENT

TOP user-configurable defaults for options. **TOPCOLORS** color specification

BUGS

As with *ps*(1), things can change while *top* is collecting information for an update. The picture it gives is only a close approximation to reality.

SEE ALSO

kill(1), ps(1), stty(1), mem(4), renice(8)

SUNOS 5 NOTES

CPU percentage is calculated as a fraction of total available computing resources. Hence on a multiprocessor machine a single threaded process can never consume cpu time in excess of 1 divided by the number of processors. For example, on a 4 processor machine, a single threaded process will never show a cpu percentage higher than 25%. The CPU percentage column will always total approximately 100, regardless of the number of processors.

The memory summary line displays the following: "phys mem" is the total amount of physical memory that can be allocated for use by processes (it does not include memory reserved for the kernel's use), "free mem" is the amount of unallocated physical memory, "total swap" is the amount of swap area on disk that is being used, "free swap" is the amount of swap area on disk that is still available. Unlike previous versions of *top*, The swap figures will differ from the summary output of *swap*(1M) since the latter includes physical memory as well.

The column "THR" indicates the number of execution threads in the process.

In BSD Unix, process priority was represented internally as a signed offset from a zero value with an unsigned value. The "zero" value was usually something like 20, allowing for a range of priorities from -20 to 20. As implemented on SunOS 5, older versions of *top* continued to interpret process priority in this manner, even though it was no longer correct. Starting with *top* version 3.5, this was changed to agree with the rest of the system.

The SunOS 5 (Solaris 2) port was originally written by Torsten Kasch, <torsten@techfak.uni-bielefeld.de>. Many contributions have been provided by Casper Dik <Casper.Dik@sun.com>. Support for multi-cpu, calculation of CPU% and memory stats provided by Robert Boucher <boucher@sofkin.ca>, Marc Cohen <marc@aai.com>, Charles Hedrick <hedrick@geneva.rutgers.edu>, and William L. Jones <jones@chpc>.

NAME

toupper – return given string in upper case

SYNOPSIS

edrc/lib/toupper [**-h**]

toupper ["*string*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the string specified in *string* in upper case to **stdout**.

OPTIONS

-h usage message.

"*string*" string to be translated to upper case.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **tolower(3)**

NOTES

-

BUGS

-

AUTHOR

toupper was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tput_examples – examples on the use of the tput(1) command to control terminal outputs

SYNOPSIS

tput [*options*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

This man page gives examples of **tput**(1) usages.

The **tput**(1) utility uses the terminfo database to make the values of terminal-dependent capabilities and information available to the shell, to initialize, reset the terminal, set colors, font weight and much more.

OPTIONS

-

ENVIRONMENT

-

EXIT STATUS

-

FILES

-

EXAMPLES**COLOUR COMMANDS**

tput setab [1-7] # Set a background colour using ANSI escape

tput setaf [1-7] # Set a foreground colour using ANSI escape

Colors are as follows:

| Num | Colour | #define | RGB |
|-----|--------|---------|-----|
|-----|--------|---------|-----|

| | | | |
|---|---------|---------------|---------------|
| 0 | black | COLOR_BLACK | 0, 0, 0 |
| 1 | red | COLOR_RED | max, 0, 0 |
| 2 | green | COLOR_GREEN | 0, max, 0 |
| 3 | yellow | COLOR_YELLOW | max, max, 0 |
| 4 | blue | COLOR_BLUE | 0, 0, max |
| 5 | magenta | COLOR_MAGENTA | max, 0, max |
| 6 | cyan | COLOR_CYAN | 0, max, max |
| 7 | white | COLOR_WHITE | max, max, max |

There are also non-ANSI versions of the colour setting functions (setb instead of setab, and setf instead of setaf) which use different numbers, not given here.

TEXT MODE COMMANDS

```
tput bold      # Select bold mode
tput dim       # Select dim (half-bright) mode
tput smul      # Enable underline mode
tput rmul      # Disable underline mode
tput rev       # Turn on reverse video mode
tput smso      # Enter standout (bold) mode
tput rmso      # Exit standout mode
tput sgr 0     # Reset all attributes
```

CURSOR MOVEMENT COMMANDS

```
tput cup Y X # Move cursor to screen position X,Y (top left is 0,0)
tput cuf N   # Move N characters forward (right)
tput cub N   # Move N characters back (left)
tput cuu N   # Move N lines up
tput ll      # Move to last line, first column (if no cup)
tput sc      # Save the cursor position
tput rc      # Restore the cursor position
tput lines   # Output the number of lines of the terminal
tput cols    # Output the number of columns of the terminal
```

CLEAR AND INSERT COMMANDS

```
tput ech N   # Erase N characters
tput clear   # Clear screen and move the cursor to 0,0
tput el 1    # Clear to beginning of line
tput el      # clear to end of line
tput ed      # clear to end of screen
tput ich N   # Insert N characters (moves rest of line forward!)
tput il N    # Insert N lines
```

OTHER COMMANDS

```
tput bel     # play a bell
```

With compiz wobbly windows, the bel command makes the terminal wobble for a second to draw the user's attention.

EXAMPLE USAGE

```
echo "$(tput setaf 1)Red text $(tput setab 7)and white background$(tput sgr 0)"
```

PERFORMING MULTIPLE OPERATIONS AT ONCE

tput accepts scripts containing one command per line, which are executed in order before **tput** exits.

Avoid temporary files by echoing a multi-line string and piping it:

```
echo -e "setf 70etb 1" | tput -S # set fg white and bg red
```

SEE ALSO

edrcintro(1), **stty(1)**, **termcap(4)**, **terminfo(5)**, **textcolor(3)**, **tput(1)**, **tty(1)**, **tty_columns(1)**, **tty_variable(1)**

NOTES

Parts of this manpage were extracted from the blog entry on '**How to change the output color of echo in Linux**' on **stackoverflow.com** answered by Drew Noakes on January 7th 2014 and modified to fit to the **WA2L/edrc** package. See <http://stackoverflow.com/questions/5947742/how-to-change-the-output-color-of-echo-in-linux> for more information.

BUGS

-

AUTHOR

the text for this man page has its roots in the blog entry written by Drew Noakes.

tput_examples has been written by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

traceroute – print the route packets take to network host

SYNOPSIS

traceroute [**-m** max_ttl] [**-n**] [**-p** port] [**-q** nqueries] [**-r**] [**-s** src_addr] [**-t** tos] [**-w**] [**-w** wait-time] *host* [*packetsize*]

DESCRIPTION

The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one's packets follow (or finding the miscreant gateway that's discarding your packets) can be difficult. *Traceroute* utilizes the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.

The only mandatory parameter is the destination host name or IP number. The default probe datagram length is 38 bytes, but this may be increased by specifying a packet size (in bytes) after the destination host name.

Other options are:

- m** Set the max time-to-live (max number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).
- n** Print hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).
- p** Set the base UDP port number used in probes (default is 33434). Traceroute hopes that nothing is listening on UDP ports *base* to *base+nhops-1* at the destination host (so an ICMP PORT_UNREACHABLE message will be returned to terminate the route tracing). If something is listening on a port in the default range, this option can be used to pick an unused port range.
- r** Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (e.g., after the interface was dropped by *routed*(8C)).
- s** Use the following IP address (which must be given as an IP number, not a hostname) as the source address in outgoing probe packets. On hosts with more than one IP address, this option can be used to force the source address to be something other than the IP address of the interface the probe packet is sent on. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent.
- t** Set the *type-of-service* in probe packets to the following value (default zero). The value must be a decimal integer in the range 0 to 255. This option can be used to see if different types-of-service result in different paths. (If you are not running 4.4bsd, this may be academic since the normal network services like telnet and ftp don't let you control the TOS). Not all values of TOS are legal or meaningful – see the IP spec for definitions. Useful values are probably '-t 16' (low delay) and '-t 8' (high throughput).
- v** Verbose output. Received ICMP packets other than TIME_EXCEEDED and UNREACHABLEs are listed.
- w** Set the time (in seconds) to wait for a response to a probe (default 3 sec.).

This program attempts to trace the route an IP packet would follow to some internet host by launching UDP probe packets with a small ttl (time to live) then listening for an ICMP "time exceeded" reply from a gateway. We start our probes with a ttl of one and increase by one until we get an ICMP "port unreachable" (which means we got to "host") or hit a max (which defaults to 30 hops & can be changed with the **-m** flag). Three probes (change with **-q** flag) are sent at each ttl setting and a line is printed showing the ttl, address of the gateway and round trip time of each probe. If the probe answers come from different gateways, the address of each responding system will be printed. If there is no response within a 3 sec. timeout interval (changed with the **-w** flag), a "*" is printed for that probe.

We don't want the destination host to process the UDP probe packets so the destination port is set to an unlikely value (if some clod on the destination is using that value, it can be changed with the `-p` flag).

A sample use and output might be:

```
[yak 71]% traceroute nis.nsf.net.
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 56 byte packet
 1 helios.ee.lbl.gov (128.3.112.1) 19 ms 19 ms 0 ms
 2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
 3 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 39 ms 19 ms
 4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 39 ms
 5 ccn-nerif22.Berkeley.EDU (128.32.168.22) 39 ms 39 ms 39 ms
 6 128.32.197.4 (128.32.197.4) 40 ms 59 ms 59 ms
 7 131.119.2.5 (131.119.2.5) 59 ms 59 ms 59 ms
 8 129.140.70.13 (129.140.70.13) 99 ms 99 ms 80 ms
 9 129.140.71.6 (129.140.71.6) 139 ms 239 ms 319 ms
10 129.140.81.7 (129.140.81.7) 220 ms 199 ms 199 ms
11 nic.merit.edu (35.1.1.48) 239 ms 239 ms 239 ms
```

Note that lines 2 & 3 are the same. This is due to a buggy kernel on the 2nd hop system – `lbl-csam.arpa` – that forwards packets with a zero ttl (a bug in the distributed version of 4.3BSD). Note that you have to guess what path the packets are taking cross-country since the NSFNet (129.140) doesn't supply address-to-name translations for its NSSes.

A more interesting example is:

```
[yak 72]% traceroute allspice.lcs.mit.edu.
traceroute to allspice.lcs.mit.edu (18.26.0.115), 30 hops max
 1 helios.ee.lbl.gov (128.3.112.1) 0 ms 0 ms 0 ms
 2 lilac-dmc.Berkeley.EDU (128.32.216.1) 19 ms 19 ms 19 ms
 3 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 19 ms 19 ms
 4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 19 ms 39 ms 39 ms
 5 ccn-nerif22.Berkeley.EDU (128.32.168.22) 20 ms 39 ms 39 ms
 6 128.32.197.4 (128.32.197.4) 59 ms 119 ms 39 ms
 7 131.119.2.5 (131.119.2.5) 59 ms 59 ms 39 ms
 8 129.140.70.13 (129.140.70.13) 80 ms 79 ms 99 ms
 9 129.140.71.6 (129.140.71.6) 139 ms 139 ms 159 ms
10 129.140.81.7 (129.140.81.7) 199 ms 180 ms 300 ms
11 129.140.72.17 (129.140.72.17) 300 ms 239 ms 239 ms
12 * * *
13 128.121.54.72 (128.121.54.72) 259 ms 499 ms 279 ms
14 * * *
15 * * *
16 * * *
17 * * *
18 ALLSPICE.LCS.MIT.EDU (18.26.0.115) 339 ms 279 ms 279 ms
```

Note that the gateways 12, 14, 15, 16 & 17 hops away either don't send ICMP "time exceeded" messages or send them with a ttl too small to reach us. 14 – 17 are running the MIT C Gateway code that doesn't send "time exceeded"s. God only knows what's going on with 12.

The silent gateway 12 in the above may be the result of a bug in the 4.[23]BSD network code (and its derivatives): 4.x (x <= 3) sends an unreachable message using whatever ttl remains in the original datagram. Since, for gateways, the remaining ttl is zero, the ICMP "time exceeded" is guaranteed to not make it back to us. The behavior of this bug is slightly more interesting when it appears on the destination system:

```

1 helios.ee.lbl.gov (128.3.112.1) 0 ms 0 ms 0 ms
2 lilac-dmc.Berkeley.EDU (128.32.216.1) 39 ms 19 ms 39 ms
3 lilac-dmc.Berkeley.EDU (128.32.216.1) 19 ms 39 ms 19 ms
4 ccngw-ner-cc.Berkeley.EDU (128.32.136.23) 39 ms 40 ms 19 ms
5 ccn-nerif35.Berkeley.EDU (128.32.168.35) 39 ms 39 ms 39 ms
6 csgw.Berkeley.EDU (128.32.133.254) 39 ms 59 ms 39 ms
7 * * *
8 * * *
9 * * *
10 * * *
11 * * *
12 * * *
13 rip.Berkeley.EDU (128.32.131.22) 59 ms ! 39 ms ! 39 ms !

```

Notice that there are 12 "gateways" (13 is the final destination) and exactly the last half of them are "missing". What's really happening is that rip (a Sun-3 running Sun OS3.5) is using the ttl from our arriving datagram as the ttl in its ICMP reply. So, the reply will time out on the return path (with no notice sent to anyone since ICMP's aren't sent for ICMP's) until we probe with a ttl that's at least twice the path length. I.e., rip is really only 7 hops away. A reply that returns with a ttl of 1 is a clue this problem exists. Traceroute prints a "!" after the time if the ttl is ≤ 1 . Since vendors ship a lot of obsolete (DEC's Ultrix, Sun 3.x) or non-standard (HPUX) software, expect to see this problem frequently and/or take care picking the target host of your probes.

Other possible annotations after the time are **!H**, **!N**, **!P** (got a host, network or protocol unreachable, respectively), **!S** or **!F** (source route failed or fragmentation needed – neither of these should ever occur and the associated gateway is busted if you see one). If almost all the probes result in some kind of unreachable, traceroute will give up and exit.

This program is intended for use in network testing, measurement and management. It should be used primarily for manual fault isolation. Because of the load it could impose on the network, it is unwise to use *traceroute* during normal operations or from automated scripts.

AUTHOR

Implemented by Van Jacobson from a suggestion by Steve Deering. Debugged by a cast of thousands with particularly cogent suggestions or fixes from C. Philip Wood, Tim Seaver and Ken Adelman.

SEE ALSO

netstat(1), ping(8)

NAME

trash – move files to a central recycle bin (trash)

SYNOPSIS

edrc/bin/trash [**-h**]

trash -f *filename_1* { *filename_n* }

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **trash** and **untrash** command implement a trash can or recycle bin similar to the "Recycle Bin" of Microsoft Windows (TM) or the "Common Desktop Environment, CDE" (TM) as command line tools.

When a file is "trashed" with the **trash -f file** command instead of using the **rm** command, the following actions take place:

- 1.) the *file* to be trashed is copied to a configurable trash location, that, of course, should have sufficient disk space.

An economic variant of such a trash location is an automounted NFS share, so a global trash for a whole environment can be used to put trashed files and the "wasted" disk space is in general not that huge. The **trash** and **untrash** commands are designed to handle a global trash due to the fact that the trashed file is associated with a node-, MC/ServiceGuard- or similar script cluster package name. See also **trash.cfg**(4) for information about the possible configuration settings and **cltrash**(1) for information about cleaning up the trash.

- 2.) the *file* is emptied (not removed, to preserve the file handle).
- 3.) the *file* is moved to *file.TRASHED* .
- 4.) Now information about what has happened to the original *file* and how to retrieve it (with **untrash**) is written to *file.TRASHED* .

Therefore this command provides an easy and consistent way to "remove" files which, for example, cause filesystem fill ups, as database exports written to a small filesystem, and transparently document to the end user what has happened to the *file*.

This is a more reliable method then the often used practice among system administrators to move the problem causing file to a filesystem with more disk space and to create a symbolic link to it, due to the fact that an end user might not notice that the file creation 1st, caused a problem and 2nd, the created file might be incomplete/corrupt due to an export or program abort because of the filesystem fill up. Furthermore often those symlinks and especially the moved files have the tendency to get forgotten to be cleaned up (because those interventions often happen during on call times) and lots of disk space is wasted over time.

Suppose the process that caused the fill up will restart and create a new *file* with the same name. If the system administrator again trashes the *file*, then two versions of the same *file* are available for retrieval (with **untrash**).

OPTIONS

- h** usage message.
- f *filelist*** list of files to be trashed.

EXIT STATUS

- 0** no error.
- 1** LOGDIR or TRASHDIR does not exist. See **trash.cfg(4)** if you get this error.
- 2** operating system is not supported. See **osid(3)** if you get this error.
- 3** rootdir does not exist.
- 4** usage listed.
- 6** configuration file **edrc/etc/trash.cfg** does not exist.

FILES

edrc/etc/trash.cfg
configuration file for **trash**, **untrash** and **cltrash**.

edrc/var/trash/log/
default logfile location of **trash**, **untrash** and **cltrash**.

edrc/var/trash/files/
default location for trashed files.

***filename*.TRASHED**
information and instruction file to document the file trashing and to give instructions how to recover (**untrash**) the trashed file.

EXAMPLES

1) trash a file:

1.1) list files in the directory:

```
[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: ls -al

:
-rw-r----- 1 dwh_db1 dwh      17884 Jul  1 03:36 a_list.txt
-rw-r----- 1 dwh_db1 dwh  4636592 Jul  1 03:34 dwhprod.dmp
-rwxr-x--x  1 dwh_db1 dwh        500 Jul  1 03:35 export_dwhprod.sh*
-rw-r----- 1 dwh_db1 dwh        250 Jul  1 03:35 exp.par
:
```

1.2) trash the **dwhprod.dmp** file:

```
[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: trash -f dwhprod.dmp

trash files ...
      remove dwhprod.dmp ... done.
done.
```

1.3) situation in directory after invoking **trash** :

```
[ /data_dwh1/dat/exports ]
[ root@dwh_db1_prod ][ksh]: ls -al

:
-rw-r----- 1 dwh_db1 dwh      17884 Jul  1 03:36 a_list.txt
-rw-r----- 1 dwh_db1 dwh         300 Jul  1 03:42 dwhprod.dmp.TRASHE
-rwxr-x--x  1 dwh_db1 dwh         500 Jul  1 03:35 export_dwhprod.sh*
-rw-r----- 1 dwh_db1 dwh         250 Jul  1 03:35 exp.par
:
```

1.4) content of the **dwhprod.dmp.TRASHED** file:

```
#
# dwhprod.dmp.TRASHED - trash info file for trashed files
#
# [00] 01.01.2006      created by root using trash
#
```

The file:

```
-rw-r----- 1 dwh_db1 dwh    4636592 Jul  1 03:34 dwhprod.dmp
```

has been removed by administrator 'root'. To recover the file from trash, invoke:

```
untrash -f dwhprod.dmp
```

SEE ALSO

cltrash(1), **edrcintro**(1), **trash.cfg**(4), **untrash**(1)

NOTES

To resolve the node- or MC/ServiceGuard package name the **pkg_hostname**(3) command is used. Therefore the usage of the **trash** command is not restricted to systems that have MC/ServiceGuard installed.

BUGS

-

AUTHOR

trash was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

trash.cfg – configuration file for trash, untrash and cltrash

SYNOPSIS

edrc/etc/trash.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **trash** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS**TRASHDIR**

Trash basedir. This directory holds all trashed files.

Example: TRASHDIR=/dat/trash/files

Default: TRASHDIR=/var/trash/files

LOGDIR In this directory the log files of **trash(1)**, **untrash(1)** and **cltrash(1)** are saved.

Example: LOGDIR=/dat/trash/logs

Default: LOGDIR=/var/trash/logs

SEE ALSO

trash(1), **untrash(1)**, **cltrash(1)**, **edrcintro(1)**

NOTES

-

BUGS

-

AUTHOR

trash.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tusc – trace unix system calls

SYNOPSIS

tusc [options] command [args ...] | pid [pid ...]

truss [options] command | -p pid ...

DESCRIPTION

Tusc traces the system calls a process invokes and the signals it receives. It displays arguments in a symbolic way, shows the first bytes of read and write buffers as well as signal information when available. *Tusc* can attach to live processes by providing PIDs (process IDs) as argument(s).

If desired, *tusc* can be invoked as *truss*, by linking one binary to the other, in order to provide reasonable compatibility with the **UnixWare/Solaris** tracing tool (see "TRUSS COMPATIBILITY" below).

When invoked as *tusc*, the options are:

- a** Print exec arguments. If **execv(2)** syscall is being traced, print all exec arguments and their addresses.
- A** Append to output file. This option has no effect if an output file has not been specified using the **-o** option.
- B** Show branch events, on IPF (this option is not supported on PA-RISC). To obtain meaningful data, the **-i** should be used with this option.
- b bytes** Set max data size for **-r** and **-w** options. By default, when using these options, the whole read/write buffers are dumped. This option causes a maximum *bytes* number of bytes to be displayed.
- c** Count syscalls and signals. If a single **-c** is present, *tusc* will only report the number of system calls executed by the process(es), the number of errors, if any, the accumulated CPU time for each syscall, and the number of signals. If more than one **-c** is present, *tusc* will behave as if this option was not used (i.e. display its normal output) but will report these statistics upon completion. If more than two **-c**'s are present, the CPU time for each syscall instance is also reported, in seconds, between angle brackets at the beginning of each line.
- C** Is similar to **-c** option but provides more precision and more information, in particular, the low, high and average system call times.
- d [+][!]*fds*** Select file descriptors. This option allows the user to display syscalls using certain file descriptors. If the first character of the string is a plus sign (+), syscalls that do not return or whose argument list doesn't contain file descriptors are also displayed. For the remaining part of the argument string, see the **ARGUMENT LISTS** section below. Note that, unlike with the **-s** option, the system call filtering is done by *tusc*, not by the kernel. See also the **-r** and **-w** options.

For instance, using **tusc -d8 date** will only display syscalls using file descriptor 8 but using **tusc -d+8 date** will also display syscalls which do not involve file descriptors.
- e** Print environment variables. If **execv(2)** syscall is being traced, print all environment variables and their addresses.
- E** Show syscall entries. *Tusc* doesn't require OS notification for system call entry. It collects all the data it needs when the system call returns. This option will cause *tusc* to stop upon syscall entry and produce a trace. This option is only useful for processes encountering invalid syscall errors as it will show the number of the invalid syscall.
- f** Follow forks. Normally *tusc* only traces the parent side of **fork(2)**s. With this option, it will trace all children. Note that debuggers or tracers (like *tusc*) cannot be followed since only one debugger is allowed to attach to a process at any given time.

- F** Print feature level. Newer kernels provide a way to obtain the *ttrace* feature level. This option will return that level. See <sys/ttrace.h>.
- g** Do not allow attaching to any process in *tusc*'s session (see **getsid(2)**). This option is only meaningful when providing process IDs as arguments.
- G** Print syscall list (#define, id and number of arguments) for validation purposes.
- h** Print the status of the entire process hierarchy when idle. Normally, unless the **-i** is specified, *tusc* reports the waiting syscall for the last active process only. With this option the state of all processes currently traced is reported.
- i** Don't show sleeping (interruptible) syscalls. By default, if a system call takes more than 1 to 2 seconds to complete, or if the traced process is spinning in user mode, an indication of the state of the process is displayed. If this option is used, these traces are suppressed. See **ENVIRONMENT** section.

I n1[/n2]

Single-step and display instructions. This option allows the user to single-step through the target program between all or some of the system calls. The output can be fed to a disassembler to provide more information. The **n1** argument specifies that single-stepping should occur after the **n1**'th syscall. The **n2** argument specifies the number of syscalls after which single-stepping will stop (default is the remainder of the process).

The PC and instructions that are displayed (not available with older kernels) are obtained from the SIGTRAP siginfo and are the PC/instruction where the single-step originated from. If a single **-v** flag is provided, the current PC and instruction will also be displayed. If two **-v** flags are present, the register context at the time of the single-step is also printed.

- k** Keep tracing after main process exits. Normally, when specifying a binary (not a process ID) to trace and following children, *tusc* will terminate when that process exits. This option forces it to wait until all children have exited. This is especially useful when tracing daemon processes that place themselves in the background by forking and exiting the parent. When this option is used, *tusc* exits with the exit code of the process that was last traced.
- l** Print all lwpids. This option is most useful when tracing multi-threaded processes.
- n** Print process names. This option is most useful when tracing a large number of processes.

o [file|fd]

Send trace output to given file or file descriptor. Normally, *tusc*'s output goes to standard output. This option provides a way to separate *tusc*'s standard output from that of the process(es) it traces. If the argument is a valid number greater than zero, it is assumed to be a file descriptor, not a file name. For instance, "*tusc -o2*" will cause the traces to go to stderr.

- p** Print pids. By default, process IDs are not printed.

- Q** Quiet. Most informational warnings are not printed.

- r [!]**fds**** Print read buffers for given file descriptors. If this option is used, all (unless the dump size is modified using the **-b** option) read data performed by the specified file descriptor(s) is dumped. See the **ARGUMENT LISTS** section below.

The data, by default, is displayed 32 byte wide using characters followed by a blank, if the character is printable, a "C" escape sequence notation or a 2 digit hexadecimal value. See the **ENVIRONMENT** section for ways to dump data in different formats.

Note that the **-d** option has precedence: no buffer data will be displayed if the file descriptor has been selected out.

- R** Show syscall restarts. This option shows system call restarts. It is not clear how useful it is at this point.

s [!]syscalls

Select syscalls. By default, all system calls are traced. This option allows a user to trace a limited number of calls. See the **ARGUMENT LISTS** section below. The arguments can be a combination of numbers, full system call names or global expressions.

S [!]signals

Select signals. By default, all signals are traced. This option allows tracing of a limited number of signals. See the **ARGUMENT LISTS** section below. The arguments can be a combination of numbers, full signal names or signal names without the **SIG** prefix (i.e. **SIGINT** or **INT** both work). Specifying uncatchable signals has no effect.

t Detach process if it enters traced mode. If this option is used, *tusc* silently checks system call entry events and releases the process if it is performing a *ttrace(TT_PROC_SETTRC)* or a *ptrace(PT_SETTRC)* syscall. If the call is *ttrace(TT_PROC_ATTACH)* or *ptrace(PT_ATTACH)*, the process being attached to is also released if *tusc* was tracing it.

T format

Show time stamp for each syscall and signal. If the *format* argument is an empty string or if it is the string "**hires**", the time of day in *tv_sec.tv_usec* format will be displayed. Else, *format* is expected to be a string containing time formatting directives. For details, refer to the **strftime(3C)** man page.

u User thread IDs. If this option is present, user thread IDs (see pthreads) are displayed.

v Verbose. If this option is present, more verbose data is produced for some system calls. Additional **-v** options will provide more data for a subset of those syscalls (the *stat*, *statfs*, *statvfs* and *pstat* families). Three **-v** options provides the most verbosity at this time.

V Print version. This option simply prints the current *tusc* version.

w [!]fds Print write buffers for given file descriptors. Identical to **r** option but for write buffers.

x Print raw (hex) arguments. *Tusc* produces symbolic output for a large number of system calls. This option causes all arguments to be displayed in hexadecimal rather than in symbolic form. If used with the **-v** option, data that are normally displayed in symbolic form are also displayed in hexadecimal.

X Print syscall traces in an exportable format. When this option is used, *Tusc* will print the syscall traces in a format that will allow them to be imported by a spreadsheet: IDs, time stamps, syscall name, syscall argument(s) in a single parenthesized block and return value(s) are separated with tabs. Signal traces are split in 2 blocks, the first one being the signal received, the second the remainder of the data. Refer to the **ENVIRONMENT** section if another field separator is desired. All verbose or buffer data is unchanged.

y Verbose option for registers. A single **-y** will print the PC (and disassemble the instruction, on IPF), an additional **-y** will dump all registers. This option is only meaningful when **-I** is used. For backward compatibility, the **-v** option has the same effect on register display but will also produce verbose system call data where available.

z Do not print information about successful syscalls. Print only failed syscalls as well as all signals.

ARGUMENT LISTS

Several *tusc* options require a list of objects (syscalls, file descriptors, signals). The argument lists are in the form:

[!]arg1[,arg2[,...]]

The optional leading '!', means "all but". The rest of the string is a list of arguments separated by commas (',') or the *all* keyword which, except for syscall names, has the expected behavior.

The **-d** option supports multiple file descriptor ranges in the form:

[!]low-high,[low2-high2],...

The **-s** option supports a single **begin-end** range as argument:

[!]syscall-[syscall]

where syscall can be a system call number or an exact system call name. If the second value is omitted, the last system call number is implied. This is mostly useful to debug newly added syscalls internally.

SEE ALSO

ttrace(2), **ttrace_wait(2)**, **strftime(3c)**, **regcomp(3)**, **fnmatch(3)**.

A front end script, **sstep**, is available to provide on-the-fly symbolic disassembly.

EXAMPLES

tusc date Trace the **date(1)** command.

tusc 1 Trace **init(1M)**, if super-user.

tusc -f -o make.trace make

Trace **make(1)** and all the processes it invokes and send the output to the file **make.trace**.

tusc -s read,write cat /etc/passwd

Show the read and write syscalls from the command 'cat /etc/passwd'.

tusc -s !sig,mmap sleep 10

Show all system calls except **mmap** and those having 'sig' in their name.

CAVEATS

Tusc cannot trace a process being debugged or traced by another process since a process may only have one debugger at a time.

Historically, debuggers (applications using tracing interfaces) have destroyed their children, or the processes they are attached to, when exiting, or, rather, the operating system has provided this service to them. This is clearly not always a desirable behavior and *tusc* catches signals in order to detach itself from processes it may have attached to, before exiting. However, in the case of multiple tracers tracing each other and the last one attaching to a live process (for instance, "tusc tusc <PID>"), it is possible that the first tracer will exit first, taking the entire process chain with it and not allowing the last one to detach from the PID before exiting.

Newer (currently unreleased) versions of the OS allow a debugger to specify what action should be taken upon debugger exit. *Tusc* uses the **detach-on-exit** option when running on OSes that provide that option.

When tracing the crooked shell (aka C-shell, aka csh) and following children, on 11.0, one should ignore SIGUSR1 to avoid races between parent and child processes. This problem is fixed in the 11i kernel.

NOTE

The *tusc* interface and features may evolve in the future.

TRUSS COMPATIBILITY

If *tusc* is invoked as *truss*, it provide some level of compatibility with the latter:

- o The default trace output becomes stderr.
- o The default filler character is changed from dot to space.
- o The pid/lwpid display more closely resembles output from *truss*.

The following *truss* options are supported (see **truss(1)** on **UnixWare** or **Solaris** for details):

- a** Same as *tusc*'s. Compatible with *truss*.
- c** Same as *tusc*'s but only one level supported.
- d** Compatible with *truss*.
- D** Same as *tusc*'s -ccc but without the syscall summary.
- e** Same as *tusc*'s. Compatible with *truss*.
- f** Same as *tusc*'s but also turns on process ID display as it does with *truss*.
- i** Same as *tusc*'s. Compatible with *truss*.

| | |
|----------|--|
| l | Same as <i>tusc</i> 's. Compatible with <i>truss</i> . |
| o | Same as <i>tusc</i> 's. Compatible with <i>truss</i> . |
| p | Compatible with <i>truss</i> . |
| r | Same as <i>tusc</i> 's. Compatible with <i>truss</i> . |
| s | Same as <i>tusc</i> 's -S option. Compatible with <i>truss</i> . |
| t | Same as <i>tusc</i> 's -s option. Compatible with <i>truss</i> . |
| v | Argument is ignored. -v is turned on for all syscalls. |
| w | Same as <i>tusc</i> 's. Compatible with <i>truss</i> . |
| x | Argument is ignored. -x is turned on for all syscalls. |

The following *truss* options are not supported: **-m**, **-M**, **-T**, **-S**, **-u**, **-U**.

ENVIRONMENT

Tusc aligns return values, when possible, and uses dots as filler. If the **TUSCFILLER** variable is set, that string is used as filler instead if it doesn't contain tabs or non printable characters.

Unless the **-i** option is used, *tusc* uses a 2 second alarm to report sleeping system calls in the process that was last continued. The **TUSCALARM** variable may be used to change the default alarm.

Tusc has preset values to horizontally divide the output into syscall data and return value. The **TUSCMARGIN** variable may be set to specify a number of columns for the latter. Values greater than the number of columns are ignored.

The width used to print process names, when the **-n** option is used, can be adjusted by setting the **TUSC-NMW** variable to a positive number.

If the **-n** option is used, **TUSCSN** can be set to *true* to convert commands' full path names into simple names.

By default, read/write buffers contents are displayed in character pairs, using the following algorithm: ascii printable characters are printed with a space on the right, the "C" language's escape character constants (\0, \n, \r, etc...) are printed as such and all other characters are printed in hexadecimal. The **TUSCBUFMODE** variable alters this behavior. If it is set to **hex**, all words will be printed in hexadecimal. If it is set to **ascii**, printable characters will be printed without spaces on the right and non-printable characters will be shown as dots. If it is set to **dual**, hexadecimal values are printed on the left of the pipe sign, ascii on the right. In **dual** mode, output resulting from the **-r** and **-w** options is also boxed, the hex values are space separated and the columns are numbered.

Some system calls have variable argument lists. If **TUSCVARARGS** is set to *true*, these syscalls will only display the arguments examined by the OS to avoid displaying garbage. Currently, only **open(2)**, **ptrace(2)** and **sysfs(2)** have been registered as supporting variable arguments.

Tusc reports inconsistent number of arguments between what it expects and what the OS reports. Undeclared syscalls (reported as SC-### where ### is the syscall number) do not generate this warning unless **TUSCMMAW**, which stands for "mismatch argument warning", is set to **all**. Note that these warnings or any "SC-###" report indicates that *tusc*'s internal syscall table should be updated.

By default, when running in *truss* compatibility mode, unsupported options cause a warning to be printed to stderr. If **TRUSSSTRICT** is set to *true*, unsupported options will cause *truss* to exit.

Setting the **TUSCLI0N** variable to *true* will cause system error messages, time stamps and floating numbers to be localized on systems that support it.

By default, the verbose output is center justified and aligned on the colon, if the LHS is wide enough, and always includes at least 2 spaces on the left. The **TUSCVWIDTH** variable controls the position of the colon. If that variable is set to zero, the output will be left justified. The verbose data for the *getdents* syscall is always left justified, due to the nature of the output.

The **TUSCMATCH** variable controls what type of pattern matching is used. If the value is **reg**, *tusc*

evaluates regular expressions using **regcomp**(3) with the **REG_ICASE** and **REG_NOSUB** options. If the value is **sh**, *tusc* uses shell (see **sh**(1)) pattern matching. If the value is **both**, the latter is used if the former doesn't provide a match. When running in *truss* emulation mode, the variable is not evaluated and **sh** pattern matching is used.

Unless the **-x** option is used, IPv6 addresses are printed using the double colon (::) notation to skip contiguous zero elements (for instance, the IPv6 loopback would be displayed as ::1). If this behavior is not desired, setting **TUSCRAWIPV6** to **true** will cause *tusc* to print all elements.

By default, *tusc* ignores failing system calls when setting high/low execution times (**-C** option). Setting the **TUSCHILOERRS** variable to **true** will cause *tusc* to include failing syscalls.

By default, *tusc* uses tabs as field separators when the **-X** option is used. The **TUSCSEP** can be used to specify a character string (including a null string) to be used instead of the tabs.

When single-stepping on IPF, the default is to use symbolic register names. If the **TUSCPHYSREG-NAMES** variable is set to true, physical register names will be used.

Newer versions of *tusc* print user, interrupt and elapsed times upon exit when syscall statistics are requested. If the **TUSCNOUSTATS** variable is set to true, this additional information won't be printed.

By default, the **-T hires** prints the time starting at the current time. If the **TUSCTRUSSTAMPSTIMESTAMPS** variable is set to true, it will start at zero like the **-d** option of *truss* does.

AUTHOR

Written by Chris Bertin (HP).

IPF disassembler contributed by Steven M. Valentine (HP).

Copyright (c) 1996-2007 Hewlett-Packard Development Company, L.P.

DISCLAIMER

Tusc is not an official Hewlett-Packard product. As such, it is delivered "as is" with no expressed or implied warranty and is a "use at your own risk" tool.

NAME

tscat – filter to add a timestamp to the stream

SYNOPSIS

edrc/lib/tscat [**-h** | **-V**]

stream | **tscat** *options*

tscat *options* < *stream*

tail -f logfile... | **tscat** *options*

tail -f logfile -v | **tscat** *options*

tail -f logfile | **tscat -f logfile** *options*

options ::= [**-z**] [**-r**] [**-m** | **-l**] [**-n** | **-f filename**] [**-d delimiter**] [**-s session**] [[**-q**] **-a appendfile**]

AVAILABILITY

WA2L/edrc

DESCRIPTION

filter to add a timestamp and a filename to a *stream*.

OPTIONS

-h usage message.

-V print command version.

-n no file name output.

-z no output of zero length lines.

-r print raw time stamp seconds.microseconds (*s.uuuuuu*) instead of the formatted version (*YYYY-MM-DD hh:mm:s.uuuuuu*).

-m medium time stamp resolution (*s.uuuuu*).

-l low time stamp resolution, where no fraction of seconds are resolved.

-f filename (default) filename to add to *stream*.

-d delimiter
field delimiter between timestamp, filename and *stream* data.

-s *session* set session in square brackets.

-q no output to **stdout** when appending to a file.

-a *appendfile*
append timestamped output to file. When this option is specified the output to **stdout** is without timestamp.

ENVIRONMENT

\$EDRC_SESSION

set session as when using the **-s** *session* option.

EXIT STATUS

0 match

1 cannot write to the file specified in **-a** *appendfile* option.

4 usage printed.

5 version printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **log(3)**, **tee(1)**, **xlog(1)**

NOTES

For performance reasons **tscat** is implemented in C. However, for maximal compatibility the fallback for operating systems where the command (source: **edrc/src/tscat/**) has not been compiled yet it is implemented also in Perl.

BUGS

-

AUTHOR

tscat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tsize – print terminal size

SYNOPSIS

edrc/lib/tsize

AVAILABILITY

WA2L/edrc

DESCRIPTION

this command returns the current width and height of the terminal in the COLUMNS and LINES variables. This output can be used directly to initialize the environment variables using the **eval** command:

```
eval `tsize`
```

The output of **tsize** is identical to the **resize(1)** command as provided when X11 is installed. As long as **resize** is available on the system it is used internally to resolve the terminal size, if not, a fallback exists to evaluate the terminal without the use of **resize**. Therefore it is not imperative to have X11 installed on the system when using **tsize** to resolve the terminal size.

OPTIONS

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **resize(1)**, **tty(1)**,

tsize(3)

Library Commands

tsize(3)

NOTES

-

BUGS

-

AUTHOR

tsize was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tttplay – player of the tty session recorded by **tttrec**

SYNOPSIS

tttplay [*-s SPEED*] [*-n*] [*-p*] *file*

DESCRIPTION

Tttplay plays the tty session in *file*, which was recorded previously by **tttrec**(1).

When **-p** option is given, **tttplay** output the *file* as it grows. It means that you can see the "live" shell session running by another user.

If you hit any key during playback, it will go right to the next character typed. This is handy when examining sessions where a user spends a lot of time at a prompt.

Additionally, there are some special keys defined:

- + or f** double the speed of playback.
- or s** halve the speed of playback.
- 1** set playback to speed 1.0 again.

OPTIONS

- s SPEED** multiple the playing speed by *SPEED* (default is 1).
- n** no wait mode. Ignore the timing information in *file*.
- p** peek another person's tty session.

SEE ALSO

script(1), **tttrec**(1), **ttttime**(1)

NAME

`ttyrec` – a tty recorder

SYNOPSIS

ttyrec [*-a*][*-u*] [*file*]

DESCRIPTION

Ttyrec is a tty recorder. It is a derivative of **script**(1) command for recording timing information with microsecond accuracy as well. It can record emacs -nw, vi, lynx, or any programs running on tty.

Ttyrec invokes a shell and records the session until the shell exits. Recorded data can be played back with **tttyplay**(1). If the argument *file* is given, the session will be recorded in that file. Otherwise, *ttyrecord* is used as default.

OPTIONS

- a** Append the output to *file* or *ttyrecord*, rather than overwriting it.
- u** With this option, **tttyrec** automatically calls **uudecode**(1) and saves its output when uuencoded data appear on the session. It allow you to transfer files from remote host. You can call **tttyrec** with this option, login to the remote host and invoke **uuencode**(1) on it for the file you want to transfer.
- e** *command*
Invoke *command* when **tttyrec** starts.

ENVIRONMENT

SHELL If the variable *SHELL* exists, the shell forked by **tttyrec** will be that shell. If it's not set, the Bourne shell is assumed. (Most shells set this variable automatically).

SEE ALSO

script(1), **tttyplay**(1), **tttytime**(1), **uuencode**(1), **uudecode**(1)

NAME

`ttytime` – print the time of the recorded session data by `ttyrec(1)`

SYNOPSIS

`ttytime` *file...*

DESCRIPTION

Ttytime tells you the time of recorded data in seconds. For example:

```
% ttytime *.tty
173 foo.tty
1832 bar.tty
```

SEE ALSO

`script(1)`, `ttyrec(1)`, `ttypplay(1)`

NAME

tty_columns – return the current width of the terminal (tty)

SYNOPSIS

edrc/lib/tty_columns [-h]

tty_columns

AVAILABILITY

WA2L/edrc

DESCRIPTION

returns the current width of the terminal (tty).

In some circumstances the **\$COLUMN** environment variable does not contain the correct width of the terminal, **tty_columns** does.

OPTIONS

-h print usage message.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message printed.

FILES

/dev/tty terminal device file. This device file is read if the resolution of the current pseudo device of the terminal is not possible.

EXAMPLES

-

SEE ALSO

edrcintro(1), **tty_variable**(3), **tty**(1), **stty**(1)

NOTES

tty_columns calls the command **tty_variable columns** internally, to resolve the terminal width. Therefore **tty_variable**(3) can be used to replace the call to **tty_columns**.

BUGS

-

AUTHOR

tty_columns was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2014 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tty_variable – return the current width of the terminal (tty)

SYNOPSIS

edrc/lib/tty_variable [**-h**]

tty_variable *variable* [*device*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

returns the current value of a terminal (tty) setting.

This command allows to process the output of the **stty**(1) command in a cross operating system compatible format.

OPTIONS

-h print usage message.

variable tty variable value to be printed. Possible variables, among others, are: **speed**, **rows**, **columns**, **line**, **intr**, **quit**, **erase**, **kill**, **eof**, **eol**, **eol2**, **start**, **stop**, **susp**, **rprnt**, **werase**, **lnext**, **flush**, **min** and **time**.

device tty device file to be read from. If not specified the device is resolved by the **tty**(1) command. If **tty** does not return a value, the default terminal device **/dev/tty** is used.

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage message printed.

FILES

/dev/tty terminal device file. This device file is read if the resolution of the current pseudo device of the terminal is not possible.

EXAMPLES

-

SEE ALSO

edrcintro(1), **tty_columns(3)**, **tty(1)**, **stty(1)**

NOTES

-

BUGS

-

AUTHOR

tty_variable was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

tzdump – dump the contents of a zoneinfo file

SYNOPSIS

edrc/lib/tzdump *zoneinfofile*

AVAILABILITY

WA2L/edrc

DESCRIPTION

dump the contents of a **tzfile(5)** format file to stdout

OPTIONS

zoneinfofile

time zone information file.

On Linux the **tzfile(5)** are found in **/usr/share/zoneinfo/**.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

/usr/share/zoneinfo/

time zone info files on Linux.

EXAMPLES

-

SEE ALSO

edrcintro(1), **timezone(5)**, **tzfile(5)**

NOTES

-

BUGS

-

AUTHOR

tzdump was developed by an unknown author and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

uid – return the numeric user-id (UID) of a user

SYNOPSIS

edrc/lib/uid [**-h**]

uid *username*

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the numeric user-id (UID) of a given user.

If the user does not exist on the system (or in an equivalent coming from some server somewhere) an empty string is returned.

OPTIONS

-h print usage message.

username name of an existing user on the system.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 user does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **gecos(3)**, **gid(3)** **group(3)**, **homedir(3)**, **user(3)**

NOTES

-

BUGS

-

AUTHOR

uid was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.
.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

uncbz, uncbz2 – unpack a bzip2-ed cpio file(s) to current dir

SYNOPSIS

edrc/bin/uncbz [**-h**]

uncbz file...

uncbz2 [**-h**]

uncbz2 file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack bzip2ed cpio file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of bzip2-ed cpio files (**.cpio.bz2**, **.cpio.bz**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **cpio(1)**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), catcomp(1), cpio(1), bzip2(1), lscmp(1), llcomp(1), uncbz(1), uncbz2(1), ungzip(1), unrpm(1), untbz(1), untbz2(1), untgz(1)

NOTES

-

BUGS

-

AUTHOR

uncbz, uncbz2 was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

uncbz, uncbz2 – unpack a bzip2-ed cpio file(s) to current dir

SYNOPSIS

edrc/bin/uncbz [**-h**]

uncbz **file...**

uncbz2 [**-h**]

uncbz2 **file...**

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack bzip2ed cpio file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of bzip2-ed cpio files (**.cpio.bz2**, **.cpio.bz**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **cpio(1)**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **catcomp(1)**, **cpio(1)**, **bzip2(1)**, **lscomp(1)**, **lcomp(1)**, **uncbz(1)**, **uncbz2(1)**, **unzip(1)**, **unrpm(1)**, **untbz(1)**, **untbz2(1)**, **untgz(1)**

NOTES

-

BUGS

-

AUTHOR

uncbz, uncbz2 was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

uncgz – unpack a gzip-ed cpio file(s) to current dir

SYNOPSIS

edrc/bin/uncgz [-h]

uncgz file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack gzipped cpio file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of gzip-ed cpio files (**.cpio.gz**, **.cgz**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **cpio**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **catcomp(1)**, **cpio(1)**, **gzip(1)**, **lscmp(1)**, **llcomp(1)**, **uncbz(1)**, **uncbz2(1)**, **unzip(1)**, **unrpm(1)**, **untbz(1)**, **untbz2(1)**, **untgz(1)**

NOTES

-

BUGS

-

AUTHOR

uncgz was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

unzip – unpack a zipped cpio file(s) to current dir

SYNOPSIS

edrc/bin/unzip [-h]

unzip file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack zip-ed cpio file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of zip-ed cpio files (**.cpio.zip**, **.czip**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **cpio**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), catcomp(1), cpio(1), zip(1), lscmp(1), llcomp(1), uncbz(1), uncbz2(1), uncgz(1), unrpm(1), untgz(1), untbz(1), untbz2(1), unzip(1)

NOTES

-

BUGS

-

AUTHOR

unzip was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

undeb – unpack a file(s) of deb (Debian package format) file(s) to current dir

SYNOPSIS

edrc/bin/undeb [**-h**]

undeb file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack file(s) of deb file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of deb files (**.deb**).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **tar**(1).

127 the **ar** command was not found on the system.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ar(1)**, **catcomp(1)**, **cpio(1)**, **lscmp(1)**, **llcomp(1)**, **rpm2cpio(8)**, **tar(1)**, **uncbz(1)**, **uncbz2(1)**, **uncgz(1)**, **unrpm(1)**, **untbz(1)**, **untbz2(1)**, **untgz(1)**, **unzip(1)**, **zip(1)**

NOTES

-

BUGS

-

AUTHOR

undeb was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

unipath – remove duplicates in colon separated strings (\$PATH)

SYNOPSIS

edrc/lib/unipath [**-h**]

unipath *string*

AVAILABILITY

WA2L/edrc

DESCRIPTION

The main purpose of **unipath** is to remove duplicates in a colon separated strings as known from the command path (**\$PATH**) or manual path (**\$MANPATH**).

The order of the elements remains, but duplicates are removed from the list.

Therefore ' **unipath** *a:b:c:b:c* ' returns *a:b:c*.

OPTIONS

-h usage message.

string colon separated string as known from the command path (**\$PATH**).

ENVIRONMENT

-

EXIT STATUS

0 no error.

4 usage listed.

FILES

-

EXAMPLES**1) usage of unipath in a script**

The following cut-out is from a Born shell script. The usage is to tidy up the command path:

```
:
:

PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/libin:$PATH
export PATH

Scriptname=`basename $0`
Scriptpath=`dirname $0`
Basedir=`$Scriptpath/../../lib/approot`

:
:

PATH=`$Basedir/lib/unipath $Basedir/bin:$Basedir/lib:$PATH`
export PATH

:
:
```

SEE ALSO

edrcintro(1), sh(1), ksh(1), bash(1)

NOTES

-

BUGS

-

AUTHOR

unipath was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

unrpm – unpack a file(s) of rpm file(s) to current dir

SYNOPSIS

edrc/bin/unrpm [**-h**]

unrpm file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack file(s) of rpm file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of rpm files (**.rpm**).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **cpio**(1).

127 the **rpm2cpio** command was not found on the system.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **catcomp(1)**, **cpio(1)**, **zip(1)**, **lscomp(1)**, **llcomp(1)**, **rpm2cpio(8)**, **uncbz(1)**, **uncbz2(1)**, **uncgz(1)**, **untbz(1)**, **untbz2(1)**, **untgz(1)**, **unzip(1)**

NOTES

-

BUGS

-

AUTHOR

unrpm was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

untbz, untbz2 – unpack a bzip2-ed tar file(s) to current dir

SYNOPSIS

edrc/bin/untbz [**-h**]

untbz file...

untbz2 [**-h**]

untbz2 file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack bzip2ed tar file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of gzip-ed tar files (**.tar.bz**, **.tar.bz2**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **tar**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), catcomp(1), bzip2(1), lscmp(1), lcomp(1), tar(1), uncbz(1), uncbz2(1), uncgz(1), ungzip(1), untbz(1), untbz2(1), unrpm(1)

NOTES

-

BUGS

-

AUTHOR

untbz, untbz2 was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

untbz, untbz2 – unpack a bzip2-ed tar file(s) to current dir

SYNOPSIS

edrc/bin/untbz [**-h**]

untbz file...

untbz2 [**-h**]

untbz2 file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack bzip2ed tar file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of gzip-ed tar files (**.tar.bz**, **.tar.bz2**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **tar**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), catcomp(1), bzip2(1), lscmp(1), lcomp(1), tar(1), uncbz(1), uncbz2(1), uncgz(1), ungzip(1), untbz(1), untbz2(1), unrpm(1)

NOTES

-

BUGS

-

AUTHOR

untbz, untbz2 was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

untgz – unpack a gzip-ed tar file(s) to current dir

SYNOPSIS

edrc/bin/untgz [**-h**]

untgz file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack gzipped tar file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of gzip-ed tar files (**.tar.gz**, **.tgz**, etc.).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **tar**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **catcomp(1)**, **gzip(1)**, **lscomp(1)**, **llcomp(1)**, **tar(1)**, **uncbz(1)**, **uncbz2(1)**, **uncgz(1)**, **unzip(1)**, **unrpm(1)**, **untbz(1)**, **untbz2(1)**

NOTES

-

BUGS

-

AUTHOR

untgz was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2013 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

untrash – restore file(s) from trash

SYNOPSIS

edrc/bin/untrash [**-h**]

untrash -f *filename_1* { *filename_n* }

AVAILABILITY

WA2L/edrc

DESCRIPTION

restore file(s) from trash handled by the **trash**(1) command.

OPTIONS

-h usage message.

-f *filelist* list of files to be untrashed.

EXIT STATUS

- | | |
|----------|--|
| 0 | no error. |
| 1 | LOGDIR or TRASHDIR does not exist. See trash.cfg (4) if you get this error. |
| 2 | operating system is not supported. See osid (3) if you get this error. |
| 3 | rootdir does not exist. |
| 4 | usage listed. |
| 6 | configuration file edrc/etc/trash.cfg does not exist. |

FILES

edrc/etc/trash.cfg
configuration file for **trash**, **untrash** and **cltrash**.

edrc/var/trash/log/
default logfile location of **trash**, **untrash** and **cltrash**.

edrc/var/trash/files/

default location for trashed files.

filename.**TRASHED**

information and instruction file to document the file trashing and to give instructions how to recover (**untrash**) the trashed file.

EXAMPLES

-

SEE ALSO

edrcintro(1), **cltrash**(1), **trash**(1), **trash.cfg**(4)

NOTES

-

BUGS

-

AUTHOR

untrash was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

untxz – unpack a xz-ed tar file(s) to current dir

SYNOPSIS

edrc/bin/untxz [**-h**]

untxz file...

AVAILABILITY

WA2L/edrc

DESCRIPTION

unpack xz-ed tar file(s) to the current working directory.

The original file(s) remain unchanged.

OPTIONS

-h usage message.

file... list of xz-ed tar files (**.tar.xz**).

ENVIRONMENT

-

EXIT STATUS

4 usage printed

x see **tar**(1).

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **catcomp(1)**, **gzip(1)**, **lscmp(1)**, **llcomp(1)**, **tar(1)**, **uncbz(1)**, **uncbz2(1)**, **uncgz(1)**, **unzip(1)**, **unrpm(1)**, **untbz(1)**, **untbz2(1)**, **xz(1)**

NOTES

-

BUGS

-

AUTHOR

untxz was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2016 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

unzip – list, test and extract compressed files in a ZIP archive

SYNOPSIS

unzip [**-Z**] [**-cflptTuvz**[**abjnoqsCDKLMUVWX\$%^**]] *file* [.zip] [*file(s)* ...] [**-x** *xfile(s)* ...] [**-d** *exdir*]

DESCRIPTION

unzip will list, test, or extract files from a ZIP archive, commonly found on MS-DOS systems. The default behavior (with no options) is to extract into the current directory (and subdirectories below it) all files from the specified ZIP archive. A companion program, *zip*(1L), creates ZIP archives; both programs are compatible with archives created by PKWARE's *PKZIP* and *PKUNZIP* for MS-DOS, but in many cases the program options or default behaviors differ.

ARGUMENTS

file [.zip]

Path of the ZIP archive(s). If the file specification is a wildcard, each matching file is processed in an order determined by the operating system (or file system). Only the filename can be a wildcard; the path itself cannot. Wildcard expressions are similar to those supported in commonly used Unix shells (*sh*, *ksh*, *csh*) and may contain:

* matches a sequence of 0 or more characters

? matches exactly 1 character

[...] matches any single character found inside the brackets; ranges are specified by a beginning character, a hyphen, and an ending character. If an exclamation point or a caret ('!' or '^') follows the left bracket, then the range of characters within the brackets is complemented (that is, anything *except* the characters inside the brackets is considered a match). To specify a verbatim left bracket, the three-character sequence "[[]]" has to be used.

(Be sure to quote any character that might otherwise be interpreted or modified by the operating system, particularly under Unix and VMS.) If no matches are found, the specification is assumed to be a literal filename; and if that also fails, the suffix .zip is appended. Note that self-extracting ZIP files are supported, as with any other ZIP archive; just specify the .exe suffix (if any) explicitly.

[*file(s)*] An optional list of archive members to be processed, separated by spaces. (VMS versions compiled with VMSCLI defined must delimit files with commas instead. See **-v** in **OPTIONS** below.) Regular expressions (wildcards) may be used to match multiple members; see above. Again, be sure to quote expressions that would otherwise be expanded or modified by the operating system.

[**-x** *xfile(s)*]

An optional list of archive members to be excluded from processing. Since wildcard characters normally match ('/') directory separators (for exceptions see the option **-W**), this option may be used to exclude any files that are in subdirectories. For example, "unzip foo *. [ch] -x */*" would extract all C source files in the main directory, but none in any subdirectories. Without the **-x** option, all C source files in all directories within the zipfile would be extracted.

[**-d** *exdir*]

An optional directory to which to extract files. By default, all files and subdirectories are recreated in the current directory; the **-d** option allows extraction in an arbitrary directory (always assuming one has permission to write to the directory). This option need not appear at the end of the command line; it is also accepted before the zipfile specification (with the normal options), immediately after the zipfile specification, or between the *file(s)* and the **-x** option. The option and directory may be concatenated without any white space between them, but note that this may cause normal shell behavior to be suppressed. In particular, "**-d** ~" (tilde) is expanded by Unix C shells into the name of the user's home directory, but "**-d** ~" is treated as a literal subdirectory "~" of the current directory.

OPTIONS

Note that, in order to support obsolescent hardware, *unzip*'s usage screen is limited to 22 or 23 lines and should therefore be considered only a reminder of the basic *unzip* syntax rather than an exhaustive list of all possible flags. The exhaustive list follows:

- Z** *zipinfo*(1L) mode. If the first option on the command line is **-Z**, the remaining options are taken to be *zipinfo*(1L) options. See the appropriate manual page for a description of these options.
- A** [OS/2, Unix DLL] print extended help for the DLL's programming interface (API).
- c** extract files to stdout/screen ("CRT"). This option is similar to the **-p** option except that the name of each file is printed as it is extracted, the **-a** option is allowed, and ASCII-EBCDIC conversion is automatically performed if appropriate. This option is not listed in the *unzip* usage screen.
- f** freshen existing files, i.e., extract only those files that already exist on disk and that are newer than the disk copies. By default *unzip* queries before overwriting, but the **-o** option may be used to suppress the queries. Note that under many operating systems, the TZ (timezone) environment variable must be set correctly in order for **-f** and **-u** to work properly (under Unix the variable is usually set automatically). The reasons for this are somewhat subtle but have to do with the differences between DOS-format file times (always local time) and Unix-format times (always in GMT/UTC) and the necessity to compare the two. A typical TZ value is "PST8PDT" (US Pacific time with automatic adjustment for Daylight Savings Time or "summer time").
- l** list archive files (short format). The names, uncompressed file sizes and modification dates and times of the specified files are printed, along with totals for all files specified. If UnZip was compiled with OS2_EAS defined, the **-l** option also lists columns for the sizes of stored OS/2 extended attributes (EAs) and OS/2 access control lists (ACLs). In addition, the zipfile comment and individual file comments (if any) are displayed. If a file was archived from a single-case file system (for example, the old MS-DOS FAT file system) and the **-L** option was given, the filename is converted to lowercase and is prefixed with a caret (^).
- p** extract files to pipe (stdout). Nothing but the file data is sent to stdout, and the files are always extracted in binary format, just as they are stored (no conversions).
- t** test archive files. This option extracts each specified file in memory and compares the CRC (cyclic redundancy check, an enhanced checksum) of the expanded file with the original file's stored CRC value.
- T** [most OSes] set the timestamp on the archive(s) to that of the newest file in each one. This corresponds to *zip*'s **-go** option except that it can be used on wildcard zipfiles (e.g., "*unzip -T *.zip*") and is much faster.
- u** update existing files and create new ones if needed. This option performs the same function as the **-f** option, extracting (with query) files that are newer than those with the same name on disk, and in addition it extracts those files that do not already exist on disk. See **-f** above for information on setting the timezone properly.
- v** list archive files (verbose format) or show diagnostic version info. This option has evolved and now behaves as both an option and a modifier. As an option it has two purposes: when a zipfile is specified with no other options, **-v** lists archive files verbosely, adding to the basic **-l** info the compression method, compressed size, compression ratio and 32-bit CRC. In contrast to most of the competing utilities, *unzip* removes the 12 additional header bytes of encrypted entries from the compressed size numbers. Therefore, compressed size and compression ratio figures are independent of the entry's encryption status and show the correct compression performance. (The complete size of the encrypted compressed data stream for zipfile entries is reported by the more verbose *zipinfo*(1L) reports, see the separate manual.) When no zipfile is specified (that is, the complete command is simply "*unzip -v*"), a diagnostic screen is printed. In addition to the normal header with release date and version, *unzip* lists the home Info-ZIP ftp site and where to find a list of other ftp and non-ftp sites; the target operating system for which it was compiled, as well as

(possibly) the hardware on which it was compiled, the compiler and version used, and the compilation date; any special compilation options that might affect the program's operation (see also **DECRYPTION** below); and any options stored in environment variables that might do the same (see **ENVIRONMENT OPTIONS** below). As a modifier it works in conjunction with other options (e.g., **-t**) to produce more verbose or debugging output; this is not yet fully implemented but will be in future releases.

-z display only the archive comment.

MODIFIERS

-a convert text files. Ordinarily all files are extracted exactly as they are stored (as "binary" files). The **-a** option causes files identified by *zip* as text files (those with the 't' label in *zipinfo* listings, rather than 'b') to be automatically extracted as such, converting line endings, end-of-file characters and the character set itself as necessary. (For example, Unix files use line feeds (LFs) for end-of-line (EOL) and have no end-of-file (EOF) marker; Macintoshes use carriage returns (CRs) for EOLs; and most PC operating systems use CR+LF for EOLs and control-Z for EOF. In addition, IBM mainframes and the Michigan Terminal System use EBCDIC rather than the more common ASCII character set, and NT supports Unicode.) Note that *zip*'s identification of text files is by no means perfect; some "text" files may actually be binary and vice versa. *unzip* therefore prints "[text]" or "[binary]" as a visual check for each file it extracts when using the **-a** option. The **-aa** option forces all files to be extracted as text, regardless of the supposed file type. On VMS, see also **-S**.

-b [general] treat all files as binary (no text conversions). This is a shortcut for **---a**.

-b [Tandem] force the creation files with filecode type 180 ('C') when extracting Zip entries marked as "text". (On Tandem, **-a** is enabled by default, see above).

-b [VMS] auto-convert binary files (see **-a** above) to fixed-length, 512-byte record format. Doubling the option (**-bb**) forces all files to be extracted in this format. When extracting to standard output (**-c** or **-p** option in effect), the default conversion of text record delimiters is disabled for binary (**-b**) resp. all (**-bb**) files.

-B [when compiled with UNIXBACKUP defined] save a backup copy of each overwritten file. The backup file is gets the name of the target file with a tilde and optionally a unique sequence number (up to 5 digits) appended. The sequence number is applied whenever another file with the original name plus tilde already exists. When used together with the "overwrite all" option **-o**, numbered backup files are never created. In this case, all backup files are named as the original file with an appended tilde, existing backup files are deleted without notice. This feature works similarly to the default behavior of *emacs*(1) in many locations.

Example: the old copy of "foo" is renamed to "foo~".

Warning: Users should be aware that the **-B** option does not prevent loss of existing data under all circumstances. For example, when *unzip* is run in overwrite-all mode, an existing "foo~" file is deleted before *unzip* attempts to rename "foo" to "foo~". When this rename attempt fails (because of a file locks, insufficient privileges, or ...), the extraction of "foo~" gets cancelled, but the old backup file is already lost. A similar scenario takes place when the sequence number range for numbered backup files gets exhausted (99999, or 65535 for 16-bit systems). In this case, the backup file with the maximum sequence number is deleted and replaced by the new backup version without notice.

-C use case-insensitive matching for the selection of archive entries from the command-line list of extract selection patterns. *unzip*'s philosophy is "you get what you ask for" (this is also responsible for the **-L/-U** change; see the relevant options below). Because some file systems are fully case-sensitive (notably those under the Unix operating system) and because both ZIP archives and *unzip* itself are portable across platforms, *unzip*'s default behavior is to match both wildcard and literal filenames case-sensitively. That is, specifying "makefile" on the command line will *only* match "makefile" in the archive, not "Makefile" or "MAKEFILE" (and similarly for wildcard specifications). Since this does not correspond to the behavior of many other operating/file

systems (for example, OS/2 HPFS, which preserves mixed case but is not sensitive to it), the **-C** option may be used to force all filename matches to be case-insensitive. In the example above, all three files would then match “makefile” (or “make*”, or similar). The **-C** option affects file specs in both the normal file list and the excluded-file list (xlist).

Please note that the **-C** option does neither affect the search for the zipfile(s) nor the matching of archive entries to existing files on the extraction path. On a case-sensitive file system, *unzip* will never try to overwrite a file “FOO” when extracting an entry “foo”!

- D** skip restoration of timestamps for extracted items. Normally, *unzip* tries to restore all meta-information for extracted items that are supplied in the Zip archive (and do not require privileges or impose a security risk). By specifying **-D**, *unzip* is told to suppress restoration of timestamps for directories explicitly created from Zip archive entries. This option only applies to ports that support setting timestamps for directories (currently ATheOS, BeOS, MacOS, OS/2, Unix, VMS, Win32, for other *unzip* ports, **-D** has no effect). The duplicated option **-DD** forces suppression of timestamp restoration for all extracted entries (files and directories). This option results in setting the timestamps for all extracted entries to the current time.

On VMS, the default setting for this option is **-D** for consistency with the behaviour of BACKUP: file timestamps are restored, timestamps of extracted directories are left at the current time. To enable restoration of directory timestamps, the negated option **--D** should be specified. On VMS, the option **-D** disables timestamp restoration for all extracted Zip archive items. (Here, a single **-D** on the command line combines with the default **-D** to do what an explicit **-DD** does on other systems.)

- E** [MacOS only] display contents of MacOS extra field during restore operation.
- F** [Acorn only] suppress removal of NFS filetype extension from stored filenames.
- F** [non-Acorn systems supporting long filenames with embedded commas, and only if compiled with ACORN_FTYPE_NFS defined] translate filetype information from ACORN RISC OS extra field blocks into a NFS filetype extension and append it to the names of the extracted files. (When the stored filename appears to already have an appended NFS filetype extension, it is replaced by the info from the extra field.)
- i** [MacOS only] ignore filenames stored in MacOS extra fields. Instead, the most compatible filename stored in the generic part of the entry’s header is used.
- j** junk paths. The archive’s directory structure is not recreated; all files are deposited in the extraction directory (by default, the current one).
- J** [BeOS only] junk file attributes. The file’s BeOS file attributes are not restored, just the file’s data.
- J** [MacOS only] ignore MacOS extra fields. All Macintosh specific info is skipped. Data-fork and resource-fork are restored as separate files.
- K** [AtheOS, BeOS, Unix only] retain SUID/SGID/Tacky file attributes. Without this flag, these attribute bits are cleared for security reasons.
- L** convert to lowercase any filename originating on an uppercase-only operating system or file system. (This was *unzip*’s default behavior in releases prior to 5.11; the new default behavior is identical to the old behavior with the **-U** option, which is now obsolete and will be removed in a future release.) Depending on the archiver, files archived under single-case file systems (VMS, old MS-DOS FAT, etc.) may be stored as all-uppercase names; this can be ugly or inconvenient when extracting to a case-preserving file system such as OS/2 HPFS or a case-sensitive one such as under Unix. By default *unzip* lists and extracts such filenames exactly as they’re stored (excepting truncation, conversion of unsupported characters, etc.); this option causes the names of all files from certain systems to be converted to lowercase. The **-LL** option forces conversion of every filename to lowercase, regardless of the originating file system.
- M** pipe all output through an internal pager similar to the Unix *more*(1) command. At the end of a screenful of output, *unzip* pauses with a “—More—” prompt; the next screenful may be viewed

by pressing the Enter (Return) key or the space bar. *unzip* can be terminated by pressing the “q” key and, on some systems, the Enter/Return key. Unlike Unix *more*(1), there is no forward-searching or editing capability. Also, *unzip* doesn’t notice if long lines wrap at the edge of the screen, effectively resulting in the printing of two or more lines and the likelihood that some text will scroll off the top of the screen before being viewed. On some systems the number of available lines on the screen is not detected, in which case *unzip* assumes the height is 24 lines.

- n never overwrite existing files. If a file already exists, skip the extraction of that file without prompting. By default *unzip* queries before extracting any file that already exists; the user may choose to overwrite only the current file, overwrite all files, skip extraction of the current file, skip extraction of all existing files, or rename the current file.
- N [Amiga] extract file comments as Amiga filenotes. File comments are created with the –c option of *zip*(1L), or with the –N option of the Amiga port of *zip*(1L), which stores filenotes as comments.
- o overwrite existing files without prompting. This is a dangerous option, so use it with care. (It is often used with –f, however, and is the only way to overwrite directory EAs under OS/2.)
- P *password*
use *password* to decrypt encrypted zipfile entries (if any). **THIS IS INSECURE!** Many multi-user operating systems provide ways for any user to see the current command line of any other user; even on stand-alone systems there is always the threat of over-the-shoulder peeking. Storing the plaintext password as part of a command line in an automated script is even worse. Whenever possible, use the non-echoing, interactive prompt to enter passwords. (And where security is truly important, use strong encryption such as Pretty Good Privacy instead of the relatively weak encryption provided by standard zipfile utilities.)
- q perform operations quietly (–qq = even quieter). Ordinarily *unzip* prints the names of the files it’s extracting or testing, the extraction methods, any file or zipfile comments that may be stored in the archive, and possibly a summary when finished with each archive. The –q[q] options suppress the printing of some or all of these messages.
- s [OS/2, NT, MS-DOS] convert spaces in filenames to underscores. Since all PC operating systems allow spaces in filenames, *unzip* by default extracts filenames with spaces intact (e.g., “EA DATA. SF”). This can be awkward, however, since MS-DOS in particular does not gracefully support spaces in filenames. Conversion of spaces to underscores can eliminate the awkwardness in some cases.
- S [VMS] convert text files (–a, –aa) into Stream_LF record format, instead of the text-file default, variable-length record format. (Stream_LF is the default record format of VMS *unzip*. It is applied unless conversion (–a, –aa and/or –b, –bb) is requested or a VMS-specific entry is processed.)
- U [UNICODE_SUPPORT only] modify or disable UTF-8 handling. When UNICODE_SUPPORT is available, the option –U forces *unzip* to escape all non-ASCII characters from UTF-8 coded filenames as “#Uxxxx” (for UCS-2 characters, or “#Lxxxxxx” for unicode codepoints needing 3 octets). This option is mainly provided for debugging purpose when the fairly new UTF-8 support is suspected to mangle up extracted filenames.

The option –UU allows to entirely disable the recognition of UTF-8 encoded filenames. The handling of filename codings within *unzip* falls back to the behaviour of previous versions.

[old, obsolete usage] leave filenames uppercase if created under MS-DOS, VMS, etc. See –L above.
- V retain (VMS) file version numbers. VMS files can be stored with a version number, in the format *file.ext;##*. By default the “;##” version numbers are stripped, but this option allows them to be retained. (On file systems that limit filenames to particularly short lengths, the version numbers may be truncated or stripped regardless of this option.)
- W [only when WILD_STOP_AT_DIR compile-time option enabled] modifies the pattern matching routine so that both ‘?’ (single-char wildcard) and ‘*’ (multi-char wildcard) do not match the

directory separator character '/'. (The two-character sequence "***" acts as a multi-char wildcard that includes the directory separator in its matched characters.) Examples:

```
"*.c" matches "foo.c" but not "mydir/foo.c"
"**.*" matches both "foo.c" and "mydir/foo.c"
"*/.*" matches "bar/foo.c" but not "baz/bar/foo.c"
"?*/.*" matches "ab/foo" and "abc/foo"
        but not "a/foo" or "a/b/foo"
```

This modified behaviour is equivalent to the pattern matching style used by the shells of some of UnZip's supported target OSs (one example is Acorn RISC OS). This option may not be available on systems where the Zip archive's internal directory separator character '/' is allowed as regular character in native operating system filenames. (Currently, UnZip uses the same pattern matching rules for both wildcard zipfile specifications and zip entry selection patterns in most ports. For systems allowing '/' as regular filename character, the -W option would not work as expected on a wildcard zipfile specification.)

- X [VMS, Unix, OS/2, NT, Tandem] restore owner/protection info (UICs and ACL entries) under VMS, or user and group info (UID/GID) under Unix, or access control lists (ACLs) under certain network-enabled versions of OS/2 (Warp Server with IBM LAN Server/Requester 3.0 to 5.0; Warp Connect with IBM Peer 1.0), or security ACLs under Windows NT. In most cases this will require special system privileges, and doubling the option (-XX) under NT instructs *unzip* to use privileges for extraction; but under Unix, for example, a user who belongs to several groups can restore files owned by any of those groups, as long as the user IDs match his or her own. Note that ordinary file attributes are always restored--this option applies only to optional, extra ownership info available on some operating systems. [NT's access control lists do not appear to be especially compatible with OS/2's, so no attempt is made at cross-platform portability of access privileges. It is not clear under what conditions this would ever be useful anyway.]
- Y [VMS] treat archived file name endings of ".nnn" (where "nnn" is a decimal number) as if they were VMS version numbers (";nnn"). (The default is to treat them as file types.) Example:
 "a.b.3" -> "a.b;3".
- \$ [MS-DOS, OS/2, NT] restore the volume label if the extraction medium is removable (e.g., a diskette). Doubling the option (-\$\$) allows fixed media (hard disks) to be labelled as well. By default, volume labels are ignored.

-I extensions

[Acorn only] overrides the extension list supplied by Unzip\$Ext environment variable. During extraction, filename extensions that match one of the items in this extension list are swapped in front of the base name of the extracted file.

- : [all but Acorn, VM/CMS, MVS, Tandem] allows to extract archive members into locations outside of the current "extraction root folder". For security reasons, *unzip* normally removes "parent dir" path components ("..") from the names of extracted file. This safety feature (new for version 5.50) prevents *unzip* from accidentally writing files to "sensitive" areas outside the active extraction folder tree head. The -: option lets *unzip* switch back to its previous, more liberal behaviour, to allow exact extraction of (older) archives that used ".." components to create multiple directory trees at the level of the current extraction folder. This option does not enable writing explicitly to the root directory ("/"). To achieve this, it is necessary to set the extraction target folder to root (e.g. -d /). However, when the -: option is specified, it is still possible to implicitly write to the root directory by specifying enough ".." path components within the zip archive. Use this option with extreme caution.
- ^ [Unix only] allow control characters in names of extracted ZIP archive entries. On Unix, a file name may contain any (8-bit) character code with the two exception '/' (directory delimiter) and NUL (0x00, the C string termination indicator), unless the specific file system has more restrictive conventions. Generally, this allows to embed ASCII control characters (or even sophisticated control sequences) in file names, at least on 'native' Unix file systems. However, it may be highly

suspicious to make use of this Unix "feature". Embedded control characters in file names might have nasty side effects when displayed on screen by some listing code without sufficient filtering. And, for ordinary users, it may be difficult to handle such file names (e.g. when trying to specify it for open, copy, move, or delete operations). Therefore, *unzip* applies a filter by default that removes potentially dangerous control characters from the extracted file names. The *-^* option allows to override this filter in the rare case that embedded filename control characters are to be intentionally restored.

- 2 [VMS] force unconditionally conversion of file names to ODS2-compatible names. The default is to exploit the destination file system, preserving case and extended file name characters on an ODS5 destination file system; and applying the ODS2-compatibility file name filtering on an ODS2 destination file system.

ENVIRONMENT OPTIONS

unzip's default behavior may be modified via options placed in an environment variable. This can be done with any option, but it is probably most useful with the *-a*, *-L*, *-C*, *-q*, *-o*, or *-n* modifiers: make *unzip* auto-convert text files by default, make it convert filenames from uppercase systems to lowercase, make it match names case-insensitively, make it quieter, or make it always overwrite or never overwrite files as it extracts them. For example, to make *unzip* act as quietly as possible, only reporting errors, one would use one of the following commands:

```
Unix Bourne shell:
    UNZIP=-qq; export UNZIP

Unix C shell:
    setenv UNZIP -qq

OS/2 or MS-DOS:
    set UNZIP=-qq

VMS (quotes for lowercase):
    define UNZIP_OPTS "-qq"
```

Environment options are, in effect, considered to be just like any other command-line options, except that they are effectively the first options on the command line. To override an environment option, one may use the "minus operator" to remove it. For instance, to override one of the quiet-flags in the example above, use the command

```
unzip --q[other options] zipfile
```

The first hyphen is the normal switch character, and the second is a minus sign, acting on the *q* option. Thus the effect here is to cancel one quantum of quietness. To cancel both quiet flags, two (or more) minuses may be used:

```
unzip -t--q zipfile
unzip ---qt zipfile
```

(the two are equivalent). This may seem awkward or confusing, but it is reasonably intuitive: just ignore the first hyphen and go from there. It is also consistent with the behavior of Unix *nice*(1).

As suggested by the examples above, the default variable names are UNZIP_OPTS for VMS (where the symbol used to install *unzip* as a foreign command would otherwise be confused with the environment variable), and UNZIP for all other operating systems. For compatibility with *zip*(1L), UNZIPOPT is also accepted (don't ask). If both UNZIP and UNZIPOPT are defined, however, UNZIP takes precedence. *unzip*'s diagnostic option (*-v* with no zipfile name) can be used to check the values of all four possible *unzip* and *zipinfo* environment variables.

The timezone variable (TZ) should be set according to the local timezone in order for the *-f* and *-u* to operate correctly. See the description of *-f* above for details. This variable may also be necessary to get timestamps of extracted files to be set correctly. The WIN32 (Win9x/ME/NT4/2K/XP/2K3) port of *unzip* gets the timezone configuration from the registry, assuming it is correctly set in the Control Panel. The TZ variable is ignored for this port.

DECRYPTION

Encrypted archives are fully supported by Info-ZIP software, but due to United States export restrictions, de-/encryption support might be disabled in your compiled binary. However, since spring 2000, US export restrictions have been liberated, and our source archives do now include full crypt code. In case you need binary distributions with crypt support enabled, see the file “WHERE” in any Info-ZIP source or binary distribution for locations both inside and outside the US.

Some compiled versions of *unzip* may not support decryption. To check a version for crypt support, either attempt to test or extract an encrypted archive, or else check *unzip*’s diagnostic screen (see the `-v` option above) for “[*decryption*]” as one of the special compilation options.

As noted above, the `-P` option may be used to supply a password on the command line, but at a cost in security. The preferred decryption method is simply to extract normally; if a zipfile member is encrypted, *unzip* will prompt for the password without echoing what is typed. *unzip* continues to use the same password as long as it appears to be valid, by testing a 12-byte header on each file. The correct password will always check out against the header, but there is a 1-in-256 chance that an incorrect password will as well. (This is a security feature of the PKWARE zipfile format; it helps prevent brute-force attacks that might otherwise gain a large speed advantage by testing only the header.) In the case that an incorrect password is given but it passes the header test anyway, either an incorrect CRC will be generated for the extracted data or else *unzip* will fail during the extraction because the “decrypted” bytes do not constitute a valid compressed data stream.

If the first password fails the header check on some file, *unzip* will prompt for another password, and so on until all files are extracted. If a password is not known, entering a null password (that is, just a carriage return or “Enter”) is taken as a signal to skip all further prompting. Only unencrypted files in the archive(s) will thereafter be extracted. (In fact, that’s not quite true; older versions of *zip*(1L) and *zipcloak*(1L) allowed null passwords, so *unzip* checks each encrypted file to see if the null password works. This may result in “false positives” and extraction errors, as noted above.)

Archives encrypted with 8-bit passwords (for example, passwords with accented European characters) may not be portable across systems and/or other archivers. This problem stems from the use of multiple encoding methods for such characters, including Latin-1 (ISO 8859-1) and OEM code page 850. DOS *PKZIP* 2.04g uses the OEM code page; Windows *PKZIP* 2.50 uses Latin-1 (and is therefore incompatible with DOS *PKZIP*); Info-ZIP uses the OEM code page on DOS, OS/2 and Win3.x ports but ISO coding (Latin-1 etc.) everywhere else; and Nico Mak’s *WinZip* 6.x does not allow 8-bit passwords at all. *UnZip* 5.3 (or newer) attempts to use the default character set first (e.g., Latin-1), followed by the alternate one (e.g., OEM code page) to test passwords. On EBCDIC systems, if both of these fail, EBCDIC encoding will be tested as a last resort. (EBCDIC is not tested on non-EBCDIC systems, because there are no known archivers that encrypt using EBCDIC encoding.) ISO character encodings other than Latin-1 are not supported. The new addition of (partially) Unicode (resp. UTF-8) support in *UnZip* 6.0 has not yet been adapted to the encryption password handling in *unzip*. On systems that use UTF-8 as native character encoding, *unzip* simply tries decryption with the native UTF-8 encoded password; the built-in attempts to check the password in translated encoding have not yet been adapted for UTF-8 support and will consequently fail.

EXAMPLES

To use *unzip* to extract all members of the archive *letters.zip* into the current directory and subdirectories below it, creating any subdirectories as necessary:

```
unzip letters
```

To extract all members of *letters.zip* into the current directory only:

```
unzip -j letters
```

To test *letters.zip*, printing only a summary message indicating whether the archive is OK or not:

```
unzip -tq letters
```

To test *all* zipfiles in the current directory, printing only the summaries:

```
unzip -tq \*.zip
```

(The backslash before the asterisk is only required if the shell expands wildcards, as in Unix; double quotes could have been used instead, as in the source examples below.) To extract to standard output all members of *letters.zip* whose names end in *.tex*, auto-converting to the local end-of-line convention and piping the output into *more*(1):

```
unzip -ca letters \*.tex | more
```

To extract the binary file *paper1.dvi* to standard output and pipe it to a printing program:

```
unzip -p articles paper1.dvi | dvips
```

To extract all FORTRAN and C source files--*.f, *.c, *.h, and Makefile--into the /tmp directory:

```
unzip source.zip "[fch]" Makefile -d /tmp
```

(the double quotes are necessary only in Unix and only if globbing is turned on). To extract all FORTRAN and C source files, regardless of case (e.g., both *.c and *.C, and any makefile, Makefile, MAKEFILE or similar):

```
unzip -C source.zip "[fch]" makefile -d /tmp
```

To extract any such files but convert any uppercase MS-DOS or VMS names to lowercase and convert the line-endings of all of the files to the local standard (without respect to any files that might be marked "binary"):

```
unzip -aaCL source.zip "[fch]" makefile -d /tmp
```

To extract only newer versions of the files already in the current directory, without querying (NOTE: be careful of unzipping in one timezone a zipfile created in another--ZIP archives other than those created by Zip 2.1 or later contain no timezone information, and a "newer" file from an eastern timezone may, in fact, be older):

```
unzip -fo sources
```

To extract newer versions of the files already in the current directory and to create any files not already there (same caveat as previous example):

```
unzip -uo sources
```

To display a diagnostic screen showing which *unzip* and *zipinfo* options are stored in environment variables, whether decryption support was compiled in, the compiler with which *unzip* was compiled, etc.:

```
unzip -v
```

In the last five examples, assume that UNZIP or UNZIP_OPTS is set to -q. To do a singly quiet listing:

```
unzip -l file.zip
```

To do a doubly quiet listing:

```
unzip -ql file.zip
```

(Note that the ".zip" is generally not necessary.) To do a standard listing:

```
unzip --ql file.zip
```

or

```
unzip -l-q file.zip
```

or

```
unzip -l--q file.zip
```

(Extra minuses in options don't hurt.)

TIPS

The current maintainer, being a lazy sort, finds it very useful to define a pair of aliases: *tt* for "unzip -tq" and *ii* for "unzip -Z" (or "zipinfo"). One may then simply type "*tt* zipfile" to test an archive, something that is worth making a habit of doing. With luck *unzip* will report "No errors detected in compressed data of zipfile.zip," after which one may breathe a sigh of relief.

The maintainer also finds it useful to set the UNZIP environment variable to “-aL” and is tempted to add “-C” as well. His ZIPINFO variable is set to “-z”.

DIAGNOSTICS

The exit status (or error level) approximates the exit codes defined by PKWARE and takes on the following values, except under VMS:

- | | |
|----|---|
| 0 | normal; no errors or warnings detected. |
| 1 | one or more warning errors were encountered, but processing completed successfully anyway. This includes zipfiles where one or more files was skipped due to unsupported compression method or encryption with an unknown password. |
| 2 | a generic error in the zipfile format was detected. Processing may have completed successfully anyway; some broken zipfiles created by other archivers have simple work-arounds. |
| 3 | a severe error in the zipfile format was detected. Processing probably failed immediately. |
| 4 | <i>unzip</i> was unable to allocate memory for one or more buffers during program initialization. |
| 5 | <i>unzip</i> was unable to allocate memory or unable to obtain a tty to read the decryption password(s). |
| 6 | <i>unzip</i> was unable to allocate memory during decompression to disk. |
| 7 | <i>unzip</i> was unable to allocate memory during in-memory decompression. |
| 8 | [currently not used] |
| 9 | the specified zipfiles were not found. |
| 10 | invalid options were specified on the command line. |
| 11 | no matching files were found. |
| 50 | the disk is (or was) full during extraction. |
| 51 | the end of the ZIP archive was encountered prematurely. |
| 80 | the user aborted <i>unzip</i> prematurely with control-C (or similar) |
| 81 | testing or extraction of one or more files failed due to unsupported compression methods or unsupported decryption. |
| 82 | no files were found due to bad decryption password(s). (If even one file is successfully processed, however, the exit status is 1.) |

VMS interprets standard Unix (or PC) return values as other, scarier-looking things, so *unzip* instead maps them into VMS-style status codes. The current mapping is as follows: 1 (success) for normal exit, 0x7fff0001 for warning errors, and (0x7fff000? + 16*normal_unzip_exit_status) for all other errors, where the “?” is 2 (error) for *unzip* values 2, 9-11 and 80-82, and 4 (fatal error) for the remaining ones (3-8, 50, 51). In addition, there is a compilation option to expand upon this behavior: defining RETURN_CODES results in a human-readable explanation of what the error status means.

BUGS

Multi-part archives are not yet supported, except in conjunction with *zip*. (All parts must be concatenated together in order, and then “*zip -F*” (for *zip 2.x*) or “*zip -FF*” (for *zip 3.x*) must be performed on the concatenated archive in order to “fix” it. Also, *zip 3.0* and later can combine multi-part (split) archives into a combined single-file archive using “*zip -s- inarchive -O outarchive*”. See the *zip 3* manual page for more information.) This will definitely be corrected in the next major release.

Archives read from standard input are not yet supported, except with *funzip* (and then only the first member of the archive can be extracted).

Archives encrypted with 8-bit passwords (e.g., passwords with accented European characters) may not be portable across systems and/or other archivers. See the discussion in **DECRYPTION** above.

unzip's **-M** ("more") option tries to take into account automatic wrapping of long lines. However, the code may fail to detect the correct wrapping locations. First, TAB characters (and similar control sequences) are not taken into account, they are handled as ordinary printable characters. Second, depending on the actual system / OS port, *unzip* may not detect the true screen geometry but rather rely on "commonly used" default dimensions. The correct handling of tabs would require the implementation of a query for the actual tabulator setup on the output console.

Dates, times and permissions of stored directories are not restored except under Unix. (On Windows NT and successors, timestamps are now restored.)

[MS-DOS] When extracting or testing files from an archive on a defective floppy diskette, if the "Fail" option is chosen from DOS's "Abort, Retry, Fail?" message, older versions of *unzip* may hang the system, requiring a reboot. This problem appears to be fixed, but control-C (or control-Break) can still be used to terminate *unzip*.

Under DEC Ultrix, *unzip* would sometimes fail on long zipfiles (bad CRC, not always reproducible). This was apparently due either to a hardware bug (cache memory) or an operating system bug (improper handling of page faults?). Since Ultrix has been abandoned in favor of Digital Unix (OSF/1), this may not be an issue anymore.

[Unix] Unix special files such as FIFO buffers (named pipes), block devices and character devices are not restored even if they are somehow represented in the zipfile, nor are hard-linked files relinked. Basically the only file types restored by *unzip* are regular files, directories and symbolic (soft) links.

[OS/2] Extended attributes for existing directories are only updated if the **-o** ("overwrite all") option is given. This is a limitation of the operating system; because directories only have a creation time associated with them, *unzip* has no way to determine whether the stored attributes are newer or older than those on disk. In practice this may mean a two-pass approach is required: first unpack the archive normally (with or without freshening/updating existing files), then overwrite just the directory entries (e.g., "*unzip -o foo */*").

[VMS] When extracting to another directory, only the *[.foo]* syntax is accepted for the **-d** option; the simple Unix *foo* syntax is silently ignored (as is the less common VMS *foo.dir* syntax).

[VMS] When the file being extracted already exists, *unzip*'s query only allows skipping, overwriting or renaming; there should additionally be a choice for creating a new version of the file. In fact, the "overwrite" choice does create a new version; the old version is not overwritten or deleted.

SEE ALSO

funzip(1L), *zip*(1L), *zipcloak*(1L), *zipgrep*(1L), *zipinfo*(1L), *zipnote*(1L), *zipsplit*(1L)

URL

The Info-ZIP home page is currently at
<http://www.info-zip.org/pub/infozip/>
 or
<ftp://ftp.info-zip.org/pub/infozip/> .

AUTHORS

The primary Info-ZIP authors (current semi-active members of the Zip-Bugs workgroup) are: Ed Gordon (Zip, general maintenance, shared code, Zip64, Win32, Unix, Unicode); Christian Spieler (UnZip maintenance coordination, VMS, MS-DOS, Win32, shared code, general Zip and UnZip integration and optimization); Onno van der Linden (Zip); Mike White (Win32, Windows GUI, Windows DLLs); Kai Uwe Rommel (OS/2, Win32); Steven M. Schweda (VMS, Unix, support of new features); Paul Kienitz (Amiga, Win32, Unicode); Chris Herborth (BeOS, QNX, Atari); Jonathan Hudson (SMS/QDOS); Sergio Monesi (Acorn RISC OS); Harald Denker (Atari, MVS); John Bush (Solaris, Amiga); Hunter Goatley (VMS, Info-ZIP Site maintenance); Steve Salisbury (Win32); Steve Miller (Windows CE GUI), Johnny Lee (MS-DOS, Win32, Zip64); and Dave Smith (Tandem NSK).

The following people were former members of the Info-ZIP development group and provided major contributions to key parts of the current code: Greg "Cave Newt" Roelofs (UnZip, unshrink decompression); Jean-loup Gailly (deflate compression); Mark Adler (inflate decompression, fUnZip).

The author of the original unzip code upon which Info-ZIP's was based is Samuel H. Smith; Carl Mascott did the first Unix port; and David P. Kirschbaum organized and led Info-ZIP in its early days with Keith Petersen hosting the original mailing list at WSMR-SimTel20. The full list of contributors to UnZip has grown quite large; please refer to the CONTRIBS file in the UnZip source distribution for a relatively complete version.

VERSIONS

| | | |
|-------|-----------|-----------------------------------|
| v1.2 | 15 Mar 89 | Samuel H. Smith |
| v2.0 | 9 Sep 89 | Samuel H. Smith |
| v2.x | fall 1989 | many Usenet contributors |
| v3.0 | 1 May 90 | Info-ZIP (DPK, consolidator) |
| v3.1 | 15 Aug 90 | Info-ZIP (DPK, consolidator) |
| v4.0 | 1 Dec 90 | Info-ZIP (GRR, maintainer) |
| v4.1 | 12 May 91 | Info-ZIP |
| v4.2 | 20 Mar 92 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.0 | 21 Aug 92 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.01 | 15 Jan 93 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.1 | 7 Feb 94 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.11 | 2 Aug 94 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.12 | 28 Aug 94 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.2 | 30 Apr 96 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.3 | 22 Apr 97 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.31 | 31 May 97 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.32 | 3 Nov 97 | Info-ZIP (Zip-Bugs subgroup, GRR) |
| v5.4 | 28 Nov 98 | Info-ZIP (Zip-Bugs subgroup, SPC) |
| v5.41 | 16 Apr 00 | Info-ZIP (Zip-Bugs subgroup, SPC) |
| v5.42 | 14 Jan 01 | Info-ZIP (Zip-Bugs subgroup, SPC) |
| v5.5 | 17 Feb 02 | Info-ZIP (Zip-Bugs subgroup, SPC) |
| v5.51 | 22 May 04 | Info-ZIP (Zip-Bugs subgroup, SPC) |
| v5.52 | 28 Feb 05 | Info-ZIP (Zip-Bugs subgroup, SPC) |
| v6.0 | 20 Apr 09 | Info-ZIP (Zip-Bugs subgroup, SPC) |

NAME

updatedb – update a file name database

SYNOPSIS

updatedb [*options*]

DESCRIPTION

This manual page documents the GNU version of **updatedb**, which updates file name databases used by GNU **locate**. The file name databases contain lists of files that were in particular directory trees when the databases were last updated. The file name of the default database is determined when **locate** and **updatedb** are configured and installed. The frequency with which the databases are updated and the directories for which they contain entries depend on how often **updatedb** is run, and with which arguments.

In networked environments, it often makes sense to build a database at the root of each filesystem, containing the entries for that filesystem. **updatedb** is then run for each filesystem on the fileserver where that filesystem is on a local disk, to prevent thrashing the network. Users can select which databases **locate** searches using an environment variable or command line option; see **locate**(1). Databases cannot be concatenated together.

The **LOCATE02** database format was introduced in GNU findutils version 4.0 in order to allow machines with different byte orderings to share the databases. GNU **locate** can read both the old and **LOCATE02** database formats, though support for the old pre-4.0 database format will be removed shortly.

OPTIONS

--findoptions=*'-option1 -option2...'*

Global options to pass on to **find**. The environment variable **FINDOPTIONS** also sets this value. Default is none.

--localpaths=*'path1 path2...'*

Non-network directories to put in the database. Default is */*.

--netpaths=*'path1 path2...'*

Network (NFS, AFS, RFS, etc.) directories to put in the database. The environment variable **NETPATHS** also sets this value. Default is none.

--prunepaths=*'path1 path2...'*

Directories to not put in the database, which would otherwise be. Remove any trailing slashes from the path names, otherwise **updatedb** won't recognise the paths you want to omit (because it uses them as regular expression patterns). The environment variable **PRUNEPATHS** also sets this value. Default is */tmp /usr/tmp /var/tmp /afs*.

--pruneefs=*'path...'*

File systems to not put in the database, which would otherwise be. Note that files are pruned when a file system is reached; any file system mounted under an undesired file system will be ignored. The environment variable **PRUNEFS** also sets this value. Default is *nfs NFS proc*.

--output=*dbfile*

The database file to build. Default is *edrc/var/locate/locate.hostname.db*.

--localuser=*user*

The user to search non-network directories as, using **su**(1). Default is to search the non-network directories as the current user. You can also use the environment variable **LOCALUSER** to set this user.

--netuser=*user*

The user to search network directories as, using **su**(1). Default is **daemon**. You can also use the environment variable **NETUSER** to set this user.

--dbformat=*F*

Create the database in format *F*. The default format is called **LOCATE02**. Alternatively the **slocate** format is also supported. When the **slocate** format is in use, the database produced is marked as having security level 1. If you want to build a system-wide **slocate** database, you may want to

run **updatedb** as root.

--version

Print the version number of **updatedb** and exit.

--help Print a summary of the options to **updatedb** and exit.

BUGS

The **updatedb** program correctly handles filenames containing newlines, but only if the system's sort command has a working **-z** option. If you suspect that **locate** may need to return filenames containing newlines, consider using its **--null** option.

REPORTING BUGS

GNU findutils online help: <<https://www.gnu.org/software/findutils/#get-help>>

Report any translation bugs to <<https://translationproject.org/team/>>

Report any other issue via the form at the GNU Savannah bug tracker:

<<https://savannah.gnu.org/bugs/?group=findutils>>

General topics about the GNU findutils package are discussed at the *bug-findutils* mailing list:

<<https://lists.gnu.org/mailman/listinfo/bug-findutils>>

COPYRIGHT

Copyright © 1994-2022 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

find(1), **locate**(1), **xargs**(1), **locatedb**(4)

Full documentation <<https://www.gnu.org/software/findutils/updatedb>>

or available locally via: **info updatedb**

NAME

usage – print short usage or command/alias cmd

SYNOPSIS

edrc/bin/usage [-h | -t]

usage [*what*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print a short usage of additional commands and aliases available on the system which are not common on the operating system.

To print the usage of the additional commands, all ASCII files in the defined directories and the user login files **\$ENV** and **\$HOME/.cshrc** (depending on the used shell) are searched for the pattern **# usage: command ...**.

To list commands in the **usage** output, add the following line to your commands:

```
# usage: my_command [ options ]      - short command description
```

Ensure that the alignment is precisely done as in this example, otherwise the output of **usage** is not aligned and looks messy.

For binary files or not self developed commands to be listed with **usage** the convention is to add a **usage.list** file into a searched directory containing the **# usage:** entries.

Add the **[helper]** "tag" to the usage string

```
# usage: my_command [ options ]      - [helper] short command description
```

to indicate that the command is not a fully honed command, but a "helper" script which provides output or functionality which needs further manual intervention.

OPTIONS

-h usage message. The usage message also prints a template of the searched **# usage:** string that can be added to new commands.

-t print usage entry template.

what select commands starting with *what* from the whole **usage** list output.

ENVIRONMENT

\$USAGE_PATH

colon separated path of directories where to scan files for the **# usage:** pattern and to search for the **usage.list** files.

If **\$USAGE_PATH** is not set the directories **edrc/bin/**, **edrc/lib/**, **edrc/lib/ksh/**, **edrc/lib/perl/**, **edrc/lib/python/**, **edrc/lib/lua/** and **edrc/lib/daemon/** are searched by default.

EXIT STATUS

0 no error.

4 usage printed.

FILES

usage.list usage list of 3rd party commands or compiled commands.

\$HOME/.cshrc

C-shell initialization file.

\$ENV

Bourne-, Bourne-Again shell initialization file.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcman(1)**, **name(1)**, **revision(1)**, **usage.list(4)**

NOTES

-

BUGS

-

AUTHOR

usage was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

usage.list – usage definition for 3rd party commands

SYNOPSIS

edrc/bin/usage.list

edrc/lib/usage.list

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the usage definition for the **usage** command.

To avoid to edit the 3rd party commands, the usage string can be added to the **usage.list** file. This protects the work after upgrading the 3rd party command.

FILEFORMAT

The fileformat is a plain list of usage definitions as read by the **usage(1)** command.

For a description of the format, see **usage(1)**.

OPTIONS

-

EXAMPLES

```
#
# bin/usage.list - usage for compiled and 3rd party commands
#
# [00] 30.01.2003 CWa   Initial Version
# [36] 27.09.2020 CWa   +xml2csv
#

# usage: asup [ options ]           - start edrc for application support
# usage: bunzip2 options            - unzip a block-sorting .bz2 file
# usage: bzip2 options              - compresses .bz2 files to stdout
# usage: bzcat options              - decompresses .bz2 files to stdout
# usage: bzdifff options            - compare bzip2 compressed files
# usage: bzgrep options             - search possibly bzip2 compressed files for
# usage: bzip2 options              - a block-sorting file compressor to generate
# usage: bzip2recover options       - recovers data from damaged bzip2 compressed
```

```
# usage: bzmorc options      - file perusal filter for crt viewing of bzi
# usage: ccrypt options      - encrypt and decrypt files and streams
# usage: cfg2html [ options ] - collect system config information
# usage: wmic options        - Windows Management Instrumentality client
```

SEE ALSO

edrcintro(1), **usage(1)**

NOTES

The file **edrc/etc/usage.list** contains all usage strings of all commands and the **edrc/bin/usage.list** and **edrc/lib/usage.list** files.

Therefore do not edit the **edrc/etc/usage.list** file.

BUGS

-

AUTHOR

usage.list was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

`user` – return the user name for a numeric user-id (uid)

SYNOPSIS

`edrc/lib/user [-h]`

`user user-id`

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the user name for a given numeric user-id (uid).

If the user does not exist on the system (or in an equivalent coming from some server somewhere) an empty string is returned.

OPTIONS

`-h` print usage message.

`user-id` numeric user id of an existing user on the system.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 user for given *user-id* does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1) **gecos(3)**, **gid(3)**, **group(3)**, **homedir(3)**, **uid(3)**

NOTES

-

BUGS

-

AUTHOR

user was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

userclass.index – index file for userclasses read by makeuser

SYNOPSIS

\$CLASSES_DIR/index

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the index file for userclasses. This file is read by the **makeuser**(1) command to give additional information to all.

FILEFORMAT

The fileformat is a list of entries that have the format

userclass:description:company:

OPTIONS

userclass userclass, this entry has to be identical to the directory name where the files for the related userclass are stored.

description description for the *userclass*.

company Company for which the userclass is in scope. This is especially useful if more companies are supported.

EXAMPLES**1) example index file:**

```
# Templates
_template:Userclass Template, do *not* apply this userclass to a User Acc

# ACME Userclasses
acmesys:System Management Team:ACME:
acmedba:Database Administrator (Oracle Owner):ACME:

notfall:Local Emergency System User:ACME:
```



```
monitoring:Monitoring Software System User:ACME:
root:Root System User for all Operating Systems:ACME:
patrol:Patrol Software System User:ACME:

# Generic Userclasses
adba:Application DBA User Account:Customer:
user:Normal Enduser User Account:Customer:
app:Application Admin User Account (Application Owner):Customer:
ftp-login:FTP User Account (Shell login expected):Customer:
ftp-nologin:FTP User Account (Shell login not expected):Customer:
```

SEE ALSO

edrcintro(1), **makeuser(1)**

NOTES

-

BUGS

-

AUTHOR

userclass.index was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

user_info – print user information

SYNOPSIS

edrc/bin/user_info [**-h**]

user_info [**-u** *username*] [**-c** *configfile*] [**-R**]

user_info **-F** *FIELDNAME* [**-u** *username*] [**-c** *configfile*]

user_info **-L**

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **user_info** command is used to display all information for a given user. With this command a single interface is given to query all permissions granted to a user. Therefore it is not needed to query each file that represent a certain permission (passwd, ftpusers, ...) separately.

OPTIONS

” When invoking **user_info** without an option, all information/permissions of a user are printed. If the *root* user starts the **user_info** command, all permissions are printed, if a non-root user uses **user_info** only the permissions that are readable by that user are printed. Therefore **user_info** does not add an extra security level, all information that would be possible to be read by a user accessing the information directly (as viewing the */etc/passwd* file) are accessible thru **user_info**, too.

-h usage message.

-n *username*
The user whose permissions to display.

-R Print the complete user record (all fields) in **FIELDNAME='value'** format. This output is intended to be used for further data processing when there is a need to compute all fields of a user. This is faster then accessing all fields sequentially using the **-F** option. To display all possible fields, use the **-L**

-F *FIELDNAME*
Query the value for the given **FIELDNAME**.

-L Print the complete list of all available fields including a short description of the field.

-c *configfile*

Specify a configfile deviating from the default *edrc/etc/user_info.cfg*.

ENVIRONMENT

-

EXIT STATUS

- | | |
|-----------|--|
| 0 | no error. |
| 2 | operating system is not supported, yet. See osid (3) if you get this error. |
| 4 | usage printed. |
| 6 | the configuration file user_info.cfg does not exist. |
| 11 | temporary directory could not be claimed or created in /tmp . Check the system temporary directory /tmp if you get this error, it is an indicator of system intrusion. |

FILES

etc/user_info.cfg

configuration file of **user_info**, see **user_info.cfg**(4) for more information.

EXAMPLES

-

SEE ALSO

edrcintro(1), **user_info.cfg**(4)

NOTES

-

BUGS

If a non-root user issues "**user_info -u** *username -F FIELDNAME* " and the *FIELDNAME* is a field where the calling user has no access to the information (as to the encrypted password) **user_info** returns an empty string without informing, that the calling user has no permission to view the value of that field.

user_info is currently completely implemented for HP-UX only.

AUTHOR

user_info was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

user_info.cfg – configuration file for user_info

SYNOPSIS

edrc/etc/user_info.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **user_info** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

PASSWD User account definition file.

Example: PASSWD=/etc/passwd

Default: PASSWD=/etc/passwd

SHADOW

Encrypted password file.

Example: SHADOW=/etc/shadow

Default: SHADOW=/etc/shadow

USERS_MAP

SAMBA user map.

Example: USERS_MAP=/etc/opt/samba/users.map

Default: USERS_MAP=/etc/opt/samba/users.map

FTPUSERS

Ftpusers. Users which are denied from executing FTP to a host.

Example: FTPUSERS=/etc/ftpd/ftpusers

Default: FTPUSERS=/etc/ftpd/ftpusers

GROUP Group definitions.

Example: GROUP=/etc/group

Default: GROUP=/etc/group

NOAGE_EXCLUDE

Users whose passwords are not aged, but who are allowed to do a direct login.

Example: NOAGE_EXCLUDE=/etc/user-db/noage_exclude

Default: NOAGE_EXCLUDE=/opt/EDS/data/exclude

TCB_AUTH

Authentication basedir of a HP trusted system.

Example: TCB_AUTH=/tcb/files/auth

Default: TCB_AUTH=/tcb/files/auth

SU_ALLOW

Allow switch user file.

Example: SU_ALLOW=/etc/user-db/su.allow

Default: SU_ALLOW=/etc/su.allow

SU_ANY Any user "user". This user is a pseudonym for every user and should not be defined on the system and is used to allow a switch user to any user in SU_ALLOW.

Example: SU_ANY=any

Default: SU_ANY=any

GENUSER_OWNERMAP

Ownership of general application users.

Example: GENUSER_OWNERMAP=/etc/user-db/passwd.ownermap

Default: GENUSER_OWNERMAP=/etc/passwd.ownermap

GENUSER_NO_OWNER

Not owned "user". This user is a pseudonym if a generic user is used but has no owner currently. This user should not be defined on the system.

Example: GENUSER_NO_OWNER=no_owner

Default: GENUSER_NO_OWNER=no_owner

GENUSER_NOT_USED

Not used "user". This user is a pseudonym if a generic user is created on the system but if this user is not used currently.

Example: GENUSER_NOT_USED=not_used

Default: GENUSER_NOT_USED=not_used

CRON_ALLOW

Local system file that contains a list of users which are allowed to define crontab entries.

Example: CRON_ALLOW=/usr/lib/cron/cron.allow

Default: CRON_ALLOW=/usr/lib/cron/cron.allow

AT_ALLOW

System file that contains a list of users which are allowed to define own at jobs.

Example: AT_ALLOW=/usr/lib/cron/at.allow

Default: AT_ALLOW=/usr/lib/cron/at.allow

CRON_ALLOW_GLOBAL

Global file holding a list of users which are allowed to define crontab entries under certain circumstances. This file is normally used as input for a user creation script writing the selected users to the **CRON_ALLOW** file.

Example: CRON_ALLOW_GLOBAL=/etc/user-db/cron.allow.global

Default: CRON_ALLOW_GLOBAL=/usr/lib/cron/cron.allow.global

AT_ALLOW_GLOBAL

Global file holding a list of users which are allowed to define crontab entries under certain circumstances. This file is normally used as input for a user creation script writing the selected users to the CRON_ALLOW file.

Example: AT_ALLOW_GLOBAL=/etc/user-db/at.allow.global

Default: AT_ALLOW_GLOBAL=/usr/lib/cron/at.allow.global

ROLE_MAP

File which holds the roles a user is assigned to.

Example: ROLE_MAP=/etc/user-db/passwd.rolemap

Default: ROLE_MAP=/etc/passwd.rolemap

SMBACCESSLOG

File containing the last samba access for a user.

Example: SMBACCESSLOG=/opt/ACME/log/smbaccess.log

Default: SMBACCESSLOG=/opt/EDS/log/smbaccess.log

SEE ALSO

edrcintro(1), **user_info(1)**

NOTES

The defaults given in this manpage are the ones that are true for the HP-UX operating system, the defaults for other operating systems might be different.

BUGS

-

AUTHOR

user_info.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

showbits – utility functions

SYNOPSIS

```
#include "utility.h"
```

```
char *showbits(unsigned int *value);
```

AVAILABILITY

WA2L/edrc

DESCRIPTION

utility.h provides utility functions.

showbits()

return a string representing the set/unset bits of the given unsigned integer *value*.

Example:

```
printf(" decimal: '%d'\n binary:  '%s'\n", 1291, showbits(1291));
```

Output:

```
decimal: '1291'  
binary:  '000000000000000000000000010100001011'
```

RETURN VALUE

-

ENVIRONMENT

-

FILES

lib/\$OSID/includes/utility.h

EXAMPLES

-

SEE ALSO

edrcintro(1), <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>, **checkopt.h(3)**, **osid(3)**, **program.h(3)**, **strings.h(3)**, **wa2lc(3)**

NOTES

-

BUGS

-

AUTHOR

utility.h was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

uuencode – decode an encoded file

SYNOPSIS

uuencode [**-flags**] [**-flag** [*value*]] [**--option-name**[[**=**] *value*]] [<file>...]

If no *file*(s) are provided, then standard input is decoded.

DESCRIPTION

uuencode transforms uuencoded files into their original form.

The encoded file(s) may be specified on the command line, or one may be read from standard input. The output file name is specified in the encoded file, but may be overridden with the **-o** option. It will have the mode of the original file, except that *setuid* and *execute* bits are not retained. If the output file is specified to be */dev/stdout* or *-*, the result will be written to standard output. If there are multiple input files and the second or subsequent file specifies standard output, the decoded data will be written to the same file as the previous output. Don't do that.

uuencode ignores any leading and trailing lines. It looks for a line that starts with "**begin**" and proceeds until the end-of-encoding marker is found. The program determines from the header line of the encoded file which of the two supported encoding schemes was used and whether or not the output file name has been encoded with base64 encoding. See *uuencode*(5).

OPTIONS

-o *file*, **--output-file**=*file*
direct output to *file*.

If specified, decoded data are written to this file. When multiple inputs are specified on the command line, this option cannot be specified. All decoded data must be written to the file name encoded in the data.

-c, **--ignore-chmod**
ignore **fchmod**(3P) errors.

By default, if the output file permissions cannot be changed to the permissions specified in the encoded data, the file will not be written out and execution stops. This option will cause that error to be ignored. The resulting file will have all the data, but the incorrect mode settings.

fchmod() errors are also ignored if **POSIXLY_CORRECT** is set in the environment. RE: <http://austingroupbugs.net/view.php?id=635>

A warning is always emitted when **fchmod**() fails.

-h, **--help**
Display usage information and exit.

-., **--more-help**
Pass the extended usage information through a pager.

-R [*cfgfile*], **--save-opts** [=*cfgfile*]
Save the option state to *cfgfile*. The default is the *last* configuration file listed in the **OPTION PRESETS** section, below. The command will exit after updating the config file.

-r *cfgfile*, **--load-opts**=*cfgfile*, **--no-load-opts**
Load options from *cfgfile*. The *no-load-opts* form will disable the loading of earlier config/rc/ini files. *--no-load-opts* is handled early, out of order.

-v [{*v*|*c*|*n* --version [{*v*|*c*|*n*}]]
Output version of program and exit. The default mode is 'v', a simple version. The 'c' mode will print copyright information and 'n' will print the full copyright notice.

OPTION PRESETS

Any option that is not marked as *not presettable* may be preset by loading values from configuration ("RC" or ".INI") file(s). The file "\$HOME/.sharrc" will be used, if present.

STANDARDS

This implementation is compliant with P1003.2b/D11.

FILES

See **OPTION PRESETS** for configuration files.

EXIT STATUS

One of the following exit values will be returned:

0 (EXIT_SUCCESS)

Successful program execution.

1 (EXIT_OPTION_ERROR)

The command options were misconfigured.

2 (EXIT_INVALID)

(warning) One or more input files contained no valid data

4 (EXIT_NO_INPUT)

(warning) The specified input file was not found

8 (EXIT_NO_OUTPUT)

The specified output file could not be created (error); or else one of the output files could not be written or its access mode could not be changed (warnings). The accompanying message(s) will distinguish.

9 (EXIT_NO_MEM)

No process memory available

66 (EX_NOINPUT)

A specified configuration file could not be loaded.

70 (EX_SOFTWARE)

libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

The exit status codes are (mostly) warning codes. As such, each code is "or"-ed into the final exit code as the input files are processed. For example, an exit code of '6' is not listed above. It is the sum of **EXIT_INVALID** and **EXIT_NO_INPUT**. It would mean that at least one input file contained invalid data and also at least one input file could not be found at all.

SEE ALSO

uuencode(1), **uuencode(4)**

AUTHORS

Free Software Foundation, Inc.

COPYRIGHT

Copyright (C) 1994-2015 Free Software Foundation, Inc. all rights reserved. This program is released under the terms of the GNU General Public License, version 3 or later.

BUGS

Please put **sharutils** in the subject line for emailed bug reports. It helps to spot the message.

If more than one *name* in the encoded files are the same, or if the second or following input files specifies standard output for the output file, then the result is probably not what is expected. Specifically, standard output will be appended to and named output files will be replaced.

Please send bug reports to: bug-gnu-utils@gnu.org

NOTES

This manual page was *AutoGen*-erated from the **uudecode** option definitions.

NAME

uuencode – encode a file into email friendly text

SYNOPSIS

uuencode [**-flags**] [**-flag** *[value]*] [**--option-name**[[=] *[value]*] [*<in-file>*] *<output-name>*

DESCRIPTION

uuencode is used to create an ASCII representation of a file that can be sent over channels that may otherwise corrupt the data. Specifically, email cannot handle binary data and will often even insert a character when the six character sequence "Orom " is seen.

uuencode will read *in-file* if provided and otherwise read data from standard in and write the encoded form to standard out. The output will begin with a header line for use by *uudecode* giving it the resulting suggested file *output-name* and access mode. If the *output-name* is specifically */dev/stdout*, then *uudecode* will emit the decoded file to standard out.

Note: *uuencode* uses buffered input and assumes that it is not hand typed from a tty. The consequence is that at a tty, you may need to hit Ctl-D several times to terminate input.

OPTIONS

-m, --base64

convert using base64.

By default, *uuencode* will encode using the traditional conversion. It is slower and less compact than base64. The encoded form of the file is expanded by 37% for UU encoding and by 35% for base64 encoding (3 bytes become 4 plus control information).

-e, --encode-file-name

encode the output file name.

Since output file names may contain characters that are not handled well by various transmission modes, you may specify that the *output-name* be base64 encoded as well. (Traditional uuencoding of the file name is not supported.)

-h, --help

Display usage information and exit.

-.!, --more-help

Pass the extended usage information through a pager.

-R *[cfgfile]*, **--save-opts** [=*cfgfile*]

Save the option state to *cfgfile*. The default is the *last* configuration file listed in the **OPTION PRESETS** section, below. The command will exit after updating the config file.

-r *cfgfile*, **--load-opts**=*cfgfile*, **--no-load-opts**

Load options from *cfgfile*. The *no-load-opts* form will disable the loading of earlier config/rc/ini files. **--no-load-opts** is handled early, out of order.

-v [{*v*|*c*|*n*} **--version** [{*v*|*c*|*n*}]]

Output version of program and exit. The default mode is 'v', a simple version. The 'c' mode will print copyright information and 'n' will print the full copyright notice.

OPTION PRESETS

Any option that is not marked as *not presettable* may be preset by loading values from configuration ("RC" or ".INI") file(s). The file "*\$HOME/.sharrc*" will be used, if present.

STANDARDS

This implementation is compliant with P1003.2b/D11.

FILES

See **OPTION PRESETS** for configuration files.

EXIT STATUS

One of the following exit values will be returned:

0 (EXIT_SUCCESS)

Successful program execution.

1 (EXIT_FAILURE)

The operation failed or the command syntax was not valid.

66 (EX_NOINPUT)

A specified configuration file could not be loaded.

70 (EX_SOFTWARE)

libopts had an internal operational error. Please report it to autogen-users@lists.sourceforge.net. Thank you.

SEE ALSO

uudecode(1), **uuencode**(4)

HISTORY

The **uuencode** command first appeared in BSD 4.0.

AUTHORS

Free Software Foundation, Inc.

COPYRIGHT

Copyright (C) 1994-2015 Free Software Foundation, Inc. all rights reserved. This program is released under the terms of the GNU General Public License, version 3 or later.

BUGS

Please put **sharutils** in the subject line for emailed bug reports. It helps to spot the message.

Please send bug reports to: bug-gnu-utils@gnu.org

NOTES

This manual page was *AutoGen*-erated from the **uuencode** option definitions.

NAME

uuencode – format of an encoded uuencode file

DESCRIPTION

Files output by uuencode(1) consist of a header line, followed by a number of body lines, and a trailer line. The uudecode(1) command will ignore any lines preceding the header or following the trailer. Lines preceding a header must not, of course, look like a header.

The header line is distinguished by having the first 5 characters be *begin* followed by a space, or else a hyphen and either *base64* or *encoded* or both (also separated with a hyphen). The *base64* option says the file has been encoded using base64. The *encoded* option says the output file *name* has been base64 encoded. It is never encoded with traditional uuencoding. This is a GNU extension. These are followed by a mode (in octal), and a string which names the remote file. The mode is separated from the *begin* clause and the file name by a single space character.

Traditional uuencoding

The traditional *uuencoded* file body consists of a number of lines, each at most 62 characters long (including the trailing newline). These consist of a character count letter, followed by the encoded characters, followed by a newline. The character count is a single printing character, and represents an integer, the number of bytes the rest of the line represents. Such integers are always in the range from 0 to 63 and can be determined by subtracting the character space (octal 40) from the character.

Groups of 3 bytes are stored in 4 characters, 6 bits per character. All are offset by a space to make the characters printing. The last line may be shorter than the normal 45 bytes. If the size is not a multiple of 3, this fact can be determined by the value of the count on the last line. Extra garbage will be included to make the character count a multiple of 4. The body is terminated by a line with a count of zero. This line consists of one ASCII space.

The trailer line consists of *end* on a line by itself.

base64 encoding

base64 encoded files follow the specified format for the body, but also include a *begin-base64* header and a trailer line of four = characters.

EXAMPLES

```
begin-base64-encoded 644 VE9ETw==
```

This introduces a base64 encoded file named, *TODO* with that name encoded using base64 encoding.

```
begin-encoded 644 5$J$3P“
```

This introduces an encoded file named, *TODO* with that name encoded using uuencoding. The encoding is a lot less friendly. Please prefer base64 encoding.

CONFORMING TO

IEEE Std 1003.1, plus extensions

The *-encoded* suffix to the *begin* header line is a GNU extension. Recipients must have the GNU *uudecode* program to decode them.

SEE ALSO

uuencode(1), uudecode(1), base64(1)

HISTORY

The *uuencode* file format appeared in BSD 4.0 .

NAME

ux2dos – convert ASCII file format from UNIX to DOS format

SYNOPSIS

edrc/lib/ux2dos [**-h**]

ux2dos *unixfile* > *dosfile*

cat *unixfile* | *ux2dos* > *dosfile*

AVAILABILITY

WA2L/edrc

DESCRIPTION

ux2dos reads the specified *unixfile* and writes it to **stdout**, converting to DOS format.

If no input file is given or if a *unixfile* is specified as -, **ux2dos** reads from **stdin**.

OPTIONS

-h print usage message.

unixfile file in UNIX file format.

dosfile file in DOS file format.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 given file does not exist.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **dos2ux(3)**

NOTES

-

BUGS

-

AUTHOR

ux2dos was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2015 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

vads – display VA (HP-Virtual Array VA-7410) Unix information

SYNOPSIS

edrc/bin/vads [-h]

vads -l lunnum

AVAILABILITY

WA2L/edrc

DESCRIPTION

display additional information of a VA-7410 of HP on HP Unix systems attached to a VirtualArray.

vads is running on osids **HP-11**, **HP-11i** and **HP-***.

OPTIONS

-h usage message.

-l lunnum calculate LUN numbers related to device names.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 operating system not supported.

4 usage printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1)

NOTES

-

BUGS

vadsp is running on **osids HP-11, HP-11i** and **HP-***.

AUTHOR

vadsp was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

volume.dat – volume definition for lots

SYNOPSIS

edrc/var/lots/objects/volume.dat
VARDIR/objects/volume.dat

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the storage volume definition for the **lots** command.

This definitions allow to save data to different locations when for instance storage capacity is shortening or due to possible conceptual considerations.

FILEFORMAT

The fileformat is a list of definitions that have the format

YEAR ; DESCRIPTION ; PATH ;

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

It is allowed to have multiple volume definitions for a year (= multiple definition lines having the same *YEAR* entry but different *PATH* definitions). The first matching *YEAR* definition will be used to save the data to, all other matching entries will be used to read and purge the data from.

OPTIONS

YEAR

to this location all data that will expire on the year specified is saved to.

If a retention exceeds the *YEAR* definitions, the data save will not be performed.

The **RETENTION_MAX** setting in the **lots.cfg** configuration file has also influence to the **save** and **lock** behaviour of **lots**. See **lots.cfg(4)**.

DESCRIPTION

description of the volume.

PATH

base path where the data is saved to. The complete path where the data is saved to is:
<PATH>/<YEAR>/<MMDD>/<DATA LIST>/<COUNTER>/data/ (example:

/dat/worm_all/2010/0426/DB_ASYPROD/000/data).

EXAMPLES

```
#
# var/lots/objects/volume.dat - data save volume destination
#
# [00] 30.09.2009 CWa   Initial Version
# [01] 29.10.2009 CWa   +2013..2019
#
#YEAR      ;DESCRIPTION                                ;PATH;
2009      ;WORM volume that expires end of 2009      ;/dat/worm_all;
2010      ;WORM volume that expires end of 2010      ;/dat/worm_all;
2011      ;WORM volume that expires end of 2011      ;/dat/worm_all;
2012      ;WORM volume that expires end of 2012      ;/dat/worm_all;
2013      ;WORM volume that expires end of 2013      ;/dat/worm_all;
2014      ;WORM volume that expires end of 2014      ;/dat/worm_all;
2015      ;WORM volume that expires end of 2015      ;/dat/worm_all;
2016      ;WORM volume that expires end of 2016      ;/dat/worm_all;
2017      ;WORM volume that expires end of 2017      ;/dat/worm_all;
2018      ;WORM volume that expires end of 2018      ;/dat/worm_all;
2019      ;WORM volume that expires end of 2019      ;/dat/worm_all;
```

SEE ALSO

edrcintro(1), **datalist.dat**(4), **schedule.dat**(4), **lots.cfg**(4), **lots**(1m)

NOTES

To verify the volume definitions, use the **lots -a list_volume** command. Only correct entries will be listed.

BUGS

-

AUTHOR

volume.dat was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2010 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

vsav – versioned file save
 vcat – cat saved file version
 vdiff – diff to saved file version
 vgrep – grep pattern in saved file version
 vlist – list all saved file versions
 vls – list last saved file versions
 vmore – more saved file version
 vpurge – purge saved file versions
 vrestore – restore saved file version

SYNOPSIS

edrc/bin/v(sav|cat|more|diff|grep|list|ls|purge|restore) [-h]

vsav *file*...

v(list|ls) *file*

v(cat|more|diff|purge|restore) *file* [*version*]

vgrep *pattern file* [*version*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The **vsav**, **vcat**, **vmore**, **vdiff**, **vgrep**, **vlist**, **vls**, **vpurge**, and **vrestore** commands enable to save, print, compare, grep, list, purge and restore *versions* of a given *file*.

When saving a *file*, a new *version* is stored, if the most recent saved *file* is not identical to the current *file*.

When restoring a *file*, automatically the *file* to be overwritten is saved.

OPTIONS

-h usage.

file file to be saved, printed, restored, compared, purged or listed.

version version of saved *file* to be restored, printed, purged or compared.

If no *version* is specified, the most recent saved *version* of the *file* is chosen.

pattern regular expression pattern.

ENVIRONMENT

\$VSAV_COMPRESS

Set to **False** if the saved file versions shall not be saved compressed.

When not set or set to **True** a saved file versions will be compressed using **gzip**(1).

EXIT STATUS

0 no error.

1 error.

4 usage printed.

FILES

edrc/var/vsav/files/

location where the saved *versions* of the *files* are stored.

edrc/var/vsav/files/<md5sum(/path/file)>-<version>[.gz]

saved *version* of *file*.

EXAMPLES

-

SEE ALSO

edrcintro(1), **gzip**(1), **sav**(1), **vvi**(1)

NOTES

-

BUGS

-

AUTHOR

vsav, vcat, vmore, vlist, vls, vrestore, vpurge, vgrep and vdiff was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2022 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

vsdfml – very simple document formatting language

SYNOPSIS

edrc/lib/vsdfml [**-h**]

vsdfml *filename* [*format*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

The purpose of **vsdfml** is to highlight documentation files on terminal output when writing recovery scripts whose purpose is to display information and not to execute commands.

To benefit from **vsdfml** you have to add it to the magic key of your information "recovery script": **#!/bin/sh vsdfml**.

Lines starting with a hash (**#**) are considered as comments and are not printed. Repetitive empty lines are reduced to one line.

All text lines beginning in the first column are set as bold font, all other text lines are set in normal font.

The text is not formatted to a certain number of columns, all text output is done as typed, this enables you to include also some ASCII graphics in the documentation.

There are some markup tags defined to set text in bold, underlined, etc. font, see section **MARKUP TAGS** for the markup tags available.

OPTIONS

-h usage message.

filename documentation file filename. When calling **vsdfml** in the magic key as shown above and in the example below, do not specify this option, it contains the filename automatically. To feed **vsdfml** with documentation text thru a pipe via **stdin**, use the pseudo file name **-**.

format Optional output format identifier. If this option is not specified, the output is formatted for manual page output (**MAN**).

TROFF the output is a man-troff format body (without header).

MAN format the output as manual page and display it using **man**.

TTY format text for terminal output.

MEDIAWIKI

the output is in MediaWiki markup.

MARKUP TAGS

Text

set *Text* in bold font.

<u>Text</u>

set *Text* in underlined font. This text is printed italic in HTML output.

<i>Text</i>

set *Text* in underlined font. This text is printed italic in HTML output.

<so>Text</so>

set *Text* in stamped out font (inverse) when using the TTY output. This text is printed bold in HTML and MAN output.

ENVIRONMENT

\$TERM terminal emulation of the current terminal. If this setting is not set correctly the whole documentation is printed, but certain font formatting (bold, underline, ...) might not be visible.

EXIT STATUS

0 no error.

4 usage listed.

11 a temporary directory could not be claimed or created in **/tmp**. Check the system temporary directory **/tmp** if you get this error, it is an indicator of system intrusion.

FILES

/tmp temporary directory.

/usr/share/terminfo

terminal capability database on systems with terminfo.

/etc/termcap

terminal capability database on systems with termcap.

EXAMPLES

1) Example of a documentation menu point.

It might makes sense to add also some **doc** tags (D:) to inform the user about the purpose of the menu point when calling the **doc** contributed command on it, see **contrib.doc(1m)**.

```
#!/usr/bin/env vsdfml
#
# 0:info_contacts - Contact Information
#
# [00] 22.05.2007 CWa    Initial Version
# [01] 03.06.2007 CWa    ++
#
# D: This is a documentation menu point, to view
# D: the documentation, start it.
#
```

PRIMARY LOCATION INFORMATION SHEET (PLIS)

Department : SMM (System Management Midrange)

Country : Switzerland

Timezone : MET

Office Hours : 08:00 - 17:00 / Monday - Friday

On Call Hours : 17:00 - 08:00 / Monday - Friday
00:00 - 24:00 / Saturday - Sunday

E-Mail Address : smm.support@acme.com
(Group)

Phone Number : +01 555 1234 5600
(during Office Hours)

Phone Number : +01 555 1234 5600
(on Call Hours)

DIRECT ACME CONTACTS

<u>Storage Management:</u>

John Doe, john.doe@acme.com, +01 555 1234 5601

Pete Miller, pete.miller@acme.com, +01 555 1234 5602

<u>Database Administration:</u>

Aaron Brown, aaron.brown@acme.com, +01 555 1234 5603

Bill Haselnut, bill.haselnut@acme.com, +01 555 1234 5604

<u>Unix System Administration:</u>

Mike Green, mike.green@acme.com, +01 555 1234 5605

Brandon Make, brandon.make@acme.com, +01 555 1234 5606

:

:

SEE ALSO

edrcintro(1), **contrib.doc(1m)**, **edrc(1m)**, **tput(1)**

NOTES

-

BUGS

-

AUTHOR

vsdfml was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

vvi – versioned file edit

SYNOPSIS

edrc/bin/vvi

vvi [*options*] [*file...*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

versioned file edit.

Prior to starting the editor, the files specified in *file...* are saved using the **vsav**(1) command internally.

This enables to use the **vls**, **vcats**, **vdiffs**, and **vreloads** commands for the edited *file*.

OPTIONS

options editor options.

file... list of files to edit.

ENVIRONMENT

\$EDITOR editor to start in **vvi** (default=**vi**).

\$VVI_MAXSAVESIZE

maximum file save size in kilo bytes. Default is **2048** kBytes.

EXIT STATUS

x exit state of **\$EDITOR**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **vcatt(1)**, **vdifff(1)**, **vi(1)**, **vls(1)**, **vmore(1)**, **vrestore(1)**, **vsav(1)**

NOTES

-

BUGS

-

AUTHOR

vvi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2024 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

wa2lc – WA2L C Programmer's Manual Overview

SYNOPSIS

```
#include "includefile.h"
```

AVAILABILITY

WA2L/edrc

DESCRIPTION

overview of custom C language include files provided in the WA2L/edrc package.

INCLUDES**C**

checkopt.h(3)

facilitate the check of correct command line option combinations.

P

program.h(3)

general program functions to ease up implement functions present in many command line commands.

S

strings.h(3)

additional string functions that are not available in **<string.h>** or **<strings.h>**.

U

utility.h(3) special utility functions.

ENVIRONMENT

-

FILES

lib/\$OSID/includes/checkopt.h

lib/\$OSID/includes/program.h

lib/\$OSID/includes/strings.h

lib/\$OSID/includes/utility.h

EXAMPLES

-

SEE ALSO

edrcintro(1), **checkopt(3)**, <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>, **checkopt.h(3)**, **osid(3)**, **program.h(3)**, **strings.h(3)**, **utility.h(3)**

NOTES

-

BUGS

Bugs and limitations are documented in the related manual pages.

AUTHOR

wa2lc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

WA2L/edrc:edrcapi – EDRC REST API

SYNOPSIS

http://server.acme.ch:9902/vn/command[?option]

https://server.acme.ch:9903/vn/command[?option]

edrc/lib/thttpd { start | stop | ... } edrcapi

edrc/bin/edrcinit { start | stop | ... } thttpd_edrcapi

edrc/bin/edrcinit { start | stop | ... } nginx_edrcapi

AVAILABILITY

WA2L/edrc

DESCRIPTION

web application **WA2L/edrc:edrcapi** (EDRC REST API) to serve WA2L/edrc information in API fashion.

/v1/hostlist

return WA2L/edrc hostlist in different formats.

See also: **hostlist(3)**.

/v1/hostlist

- *Ansible inventory json format*

/v1/hostlist?json

- *Ansible inventory json format*

/v1/hostlist?yaml

- *Ansible inventory yaml format*

/v1/hostlist?ini

- *Ansible inventory ini format*

OPTIONS

start start webserver.

stop stop webserver.

edrcapi web application name of WA2L/edrc:edrcapi.

thttpd_edrcapi

http service name of WA2L/edrc:edrcapi.

nginx_edrcapi

https service name of WA2L/edrc:edrcapi.

ENVIRONMENT

-

EXIT STATUS

-

FILES

edrc/etc/thttpd.edrcapi.cfg

configuration file for the HTTP server daemon **thttpd** which serves the application. See also **thttpd(3)** for more information.

edrc/etc/nginx.edrcapi.cfg

configuration file for the HTTPS server daemon **nginx** which serves the application. The **thttpd_edrcapi** service is a prerequisite for **nginx_edrcapi**. See also **nginx(3)** for more information.

edrc/etc/edrcinit.cfg

configuration file of the WA2L/edrc service handler. See also **edrcinit(1m)** for more information.

edrc/www/edrcapi/

base directory of WA2L/edrc:edrcapi.

edrc/www/edrcapi/.meta/

directory containing meta data for WA2L/edrc:edrcapi as application version and description.

edrc/www/edrcapi/v1/

directory containing all cgi scripts of WA2L/edrc:edrcapi with version 1.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcinit(1m)**, **edrcinit.cfg(3)**, **nginx(3)**, **nginx.edrcapi.cfg(3)**, **thttpd(3)**, **thttpd.edrcapi.cfg(3)**

NOTES

The web application is served using **thttpd**, because it is a fast, small, secure and simple HTTP server that is available and compilable on many operating systems without problems and therefore predestined for a

multi platform framework as WA2L/edrc. See **thttpd**(3) and **<http://www.acme.com/software/thttpd>** for more information.

BUGS

-

AUTHOR

WA2L/edrc:edrcapi was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

WA2L/edrc:report – THE OPERATING SYSTEM REPORT PORTAL

SYNOPSIS

http://server.acme.ch:9900

https://server.acme.ch:9901

edrc/lib/thttpd { start | stop | ... } report

edrc/bin/edrcinit { start | stop | ... } thttpd_report

edrc/bin/edrcinit { start | stop | ... } nginx_report

AVAILABILITY

WA2L/edrc

DESCRIPTION

The main purpose of the WA2L/edrc:report web application is to present operating system reports of one or more customers that are saved in a directory structure on the system where the application is running.

WA2L/edrc:report provides an easy to handle web interface for user administration. To point to customer report directories, create a symbolic link in the **/rpt** directory, to change the contents of the links displayed in the 'Links' menu, edit the file **/etc/links**, else no additional configuration is needed.

If you get the web server error **'404; Forbidden; You do not have permission to get URL from this server'** you are most likely connecting to a non-official URL of the web server. To verify this and to get the correct URL, click to the 'About' control and watch out for a red **'ensure to connect to ...'** message that informs if the application is called using a wrong URL.

See application help thru the 'Help' menu in the WA2L/edrc:report web application for more information regarding application usage.

OPTIONS

start start webserver.

stop stop webserver.

report web application name of WA2L/edrc:report.

thttpd_report
 service name of WA2L/edrc:report.

ENVIRONMENT

-

EXIT STATUS

-

FILES

edrc/etc/thttpd.report.cfg

configuration file for the HTTP server daemon **thttpd** which serves the application. See also **thttpd**(3) for more information.

edrc/etc/nginx.report.cfg

configuration file for the HTTPS server daemon **thttpd** which serves the application. The **thttpd_report** service is a prerequisite for **nginx_report**. See also **nginx**(3) for more information.

edrc/etc/edrcinit.cfg

configuration file of the WA2L/edrc service handler. See also **edrcinit**(1m) for more information.

edrc/www/report/

base directory of WA2L/edrc:report.

edrc/www/report.meta/

directory containing meta data for WA2L/edrc:report as application version and description.

edrc/www/report.htpasswd

this file holds the user password information of the WA2L/edrc:report web application. The content of this file is maintained thru the 'User Administration' menu points in the web application together with the **/etc/users** file.

When installing the WA2L/edrc package the user **admin** has the password **admin**. After the installation a personal user with the role **ADM (Admin)** should be created (with the **Users** menu) and the user **admin** should be removed.

edrc/www/report/cgi/

directory containing all cgi scripts of WA2L/edrc:report.

edrc/www/report/css/

directory containing all cascaded style sheets of WA2L/edrc:report.

edrc/www/report/errors/

error files for the **thttpd** web server of WA2L/edrc:report.

edrc/www/report/etc/

configuration directory of WA2L/edrc:report.

edrc/www/report/etc/links

configuration file for the 'Links' menu point in the WA2L/edrc:report web application. This configuration file has to be edited directly by hand.

edrc/www/report/etc/users

this file holds the user information of the WA2L/edrc:report web application. The content of this file is maintained thru the 'User Administration' menu points in the web application together with the **/.htpasswd** file.

edrc/www/report/fonts/

directory containing all additional fonts of WA2L/edrc:report.

edrc/www/report/img/

directory containing all application images of WA2L/edrc:report.

edrc/www/report/js/

directory containing all JavaScripts of WA2L/edrc:report.

edrc/www/report/main/

directory containing the static web pages of WA2L/edrc:report.

edrc/www/report/rpt/

directory containing the symbolic links to the directories containing the operating system reports.

The structure of the target report directories must be **<Customer>/<Report_Name>/<Report_File(s)>**. To add a more comprehensive description of a report, use the **duvi; chown edrc ..du_index** command in the related **<Report_Name>/** directory (use the **en:** section of the **..du_index** file).

To display and list reports in the web application, the permissions of all directories must be **drwxr-xr-x** and the permissions of the report files must be **-rw-r--r--**.

If you run a configuration, where you serve the reports in a customer's environment to a customer, but you don't want to show all reports to him/her, you can set the permissions of the related **<Report_Name>** directory you want to hide to **drwxr-x---**, sync all reports to a central reporting portal for system administration personnel (using **rsync(1)** in a script started thru a cron entry) and change the permissions on your central back to **drwxr-xr-x** (in the same script).

Example report file structure on a central report portal:

```
[ /opt/edrc/var/www/report/rpt ]
[ root@fc17mzv7t0.acme.ch ][*eshell*/bash]: ll
lrwxrwxrwx.  1 root edrc   11 Jan 12  2012 001 -> /dat/report_USA/
lrwxrwxrwx.  1 root edrc   16 Jan 12  2012 002 -> /dat/report_Switzerla
lrwxrwxrwx.  1 root edrc   10 Jan 12  2012 003 -> /dat/report_CostaRica
lrwxrwxrwx.  1 root edrc   10 Jan 12  2012 004 -> /dat/report_NewZealan
-rw-r--r--.  1 root edrc  935 Sep  1  2010 index.html
```



```
[ /opt/edrc/var/www/report/rpt/001 ]
[ root@fcl7mzv7t0.acme.ch ][*eshell*/bash]: ll
drwxr-xr-x. 2 root root 4096 Dec  2 14:00 RedrockQuarry/
drwxr-xr-x. 2 root root 4096 Dec  9 16:40 PureWater/
drwxr-xr-x. 2 root root 4096 Dec  2 15:44 SimpleFurniture/
drwxr-xr-x. 5 root root 4096 Nov  4 14:38 WA2L/
drwxr-xr-x. 2 root root 4096 Dec  8 12:54 Xenon/

[ /opt/edrc/var/www/report/rpt/001/RedrockQuarry ]
[ root@fcl7mzv7t0.acme.ch ][*eshell*/bash]: ll
drwxr-xr-x. 3 root root 4096 Dec  2 15:35 cfg2html/
drwxr-xr-x. 3 root root 4096 Nov 18 17:04 fssum/
drwxr-xr-x. 2 root root 4096 Nov  3 00:40 lots/
drwxr-xr-x. 3 root root 4096 Nov 16 19:04 performancegraph/

[ /opt/edrc/var/www/report/rpt/001/RedrockQuarry/cfg2html ]
[ root@fcl7mzv7t0.acme.ch ][*eshell*/bash]: ll
drwxr-xr-x. 2 root root 4096 Nov  3 00:50 20121103/
drwxr-xr-x. 2 root root 4096 Nov  4 00:54 20121104/
drwxr-xr-x. 2 root root 4096 Nov  5 00:52 20121105/
-rw-r-----. 1 edrc root  126 Dec  2 15:35 ..du_index

[ /opt/edrc/var/www/report/rpt/001/RedrockQuarry/cfg2html/20121103 ]
[ root@fcl7mzv7t0.acme.ch ][*eshell*/bash]: ll
-rw-r--r--. 1 root root  8983 Nov  3 00:35 pepples.redrock.com.err
-rw-r--r--. 1 root root 481265 Nov  3 00:35 pepples.redrock.com.html
-rw-r--r--. 1 root root    259 Nov  3 00:35 pepples.redrock.com.partiti
-rw-r--r--. 1 root root 448872 Nov  3 00:35 pepples.redrock.com.txt
-rw-r--r--. 1 root root   8900 Nov  3 00:40 boulder.redrock.com.err
-rw-r--r--. 1 root root 486465 Nov  3 00:40 boulder.redrock.com.html
-rw-r--r--. 1 root root    200 Nov  3 00:40 boulder.redrock.com.partiti
-rw-r--r--. 1 root root 446852 Nov  3 00:40 boulder.redrock.com.txt
```

Example report file structure in the RedrockQuarry environment, where you want to hide the **cfg2html** reports from the customer's eyes but show all other reports:

```
[ /opt/edrc/var/www/report/rpt ]
[ root@sand.redrock.com ][*eshell*/bash]: ll
lrwxrwxrwx. 1 root edrc  11 Jan 12  2012 001 -> /dat/report/
-rw-r--r--. 1 root edrc  935 Sep  1  2010 index.html

[ /opt/edrc/var/www/report/rpt/001/RedrockQuarry ]
[ root@sand.redrock.com ][*eshell*/bash]: ll
drwxr-x---. 3 root root 4096 Dec  2 15:35 cfg2html/
drwxr-xr-x. 3 root root 4096 Nov 18 17:04 fssum/
drwxr-xr-x. 2 root root 4096 Nov  3 00:40 lots/
drwxr-xr-x. 3 root root 4096 Nov 16 19:04 performancegraph/
```

edrc/www/report/tmp/

directory for temporary files generated by a cgi script. This directory is currently not used.

edrc/www/report/var/

directory containing additional static web content.

edrc/www/report/var/logos/

directory containing the factory logos of customers. The logo must have the name as returned by the **server_environment -C** command in the customer's environment and the file suffix **.png, .jpg, .jpeg** or **.gif**.

edrc/www/report/var/mylogos/

directory containing your own factory logos of customers. Files in this directory have preference over the files in **edrc/www/report/var/logos/**. Files in this directory are not touched by WA2L/edrc patch updates. The logo must have the name as returned by the **server_environment -C** command in the customer's environment and the file suffix **.png, .jpg, .jpeg** or **.gif**.

EXAMPLES

1) port setting in webserver config file **edrc/etc/tthttpd.report.cfg** of WA2L/edrc:report:

```
#
# tthttpd.report.cfg - config file for tthttpd of application 'report'
#
# [00] 07.09.2007 CWa Initial Version
#
:
port=9900
:
```

2) service definition of WA2L/edrc:report in **edrc/etc/edrcinit.cfg**:

```
#
# etc/edrcinit.cfg - configuration file for edrcinit
#
# [00] 26.05.2009 CWa Initial Version
#
:
#
# Format:
#
# <sequence>;<activated>;<guarded>;<name>;<description>;<handler>;<handl
#
:
150;Y;Y;tthttpd_report;tiny HTTP server (WA2L/edrc:report);WA2Ledrc.tthttpd
:
```

SEE ALSO

duvi(1), du_index(4), edrcinit(1m), edrcinit.cfg(3), edrcintro(1), nginx(3), nginx.report.cfg(3), tthttpd(3), tthttpd.report.cfg(3)

NOTES

The web application is served using **thttpd**, because it is a fast, small, secure and simple HTTP server that is available and compilable on many operating systems without problems and therefore predestined for a multi platform framework as WA2L/edrc. See **thttpd(3)** and **<http://www.acme.com/software/thttpd>** for more information.

BUGS

-

AUTHOR

WA2L/edrc:report was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

WA2L/edrc:shellinaboxd – shellinaboxd application

SYNOPSIS

http://server.acme.ch:8806/

https://server.acme.ch:8807/

edrc/bin/edrcinit { start | stop | ... } shellinaboxd

edrc/bin/edrcinit { start | stop | ... } nginx_shellinaboxd

AVAILABILITY

WA2L/edrc

DESCRIPTION

web application **WA2L/edrc:shellinaboxd** to serve WA2L/edrc command line over over a Web browser.

OPTIONS

start start application service.

stop stop application service.

shellinaboxd
web application name of WA2L/edrc:shellinaboxd.

shellinaboxd
http service name of WA2L/edrc:shellinaboxd.

nginx_shellinaboxd
https service name of WA2L/edrc:shellinaboxd.

ENVIRONMENT

-

EXIT STATUS

-

FILES

edrc/etc/shellinaboxd.cfg

configuration file for the HTTP server daemon **shellinaboxd** which serves the application. See also **shellinaboxd(3)** for more information.

edrc/etc/nginx.shellinaboxd.cfg

configuration file for the HTTPS server daemon **nginx** which serves the application. The **shellinaboxd** service is a prerequisite for **nginx_shellinaboxd**. See also **nginx(3)** for more information.

edrc/etc/edrcinit.cfg

configuration file of the WA2L/edrc service handler. See also **edrcinit(1m)** for more information.

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcinit(1m)**, **edrcinit.cfg(4)**, **nginx(3)**, **nginx.shellinaboxd.cfg(4)**, **shellinaboxd(3)**, **shellinaboxd.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

WA2L/edrc:shellinaboxd was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

WA2L::Util – General WA2L/edrc Utility Functions Perl Module

SYNOPSIS

Initialize environment (see also: **perlenv**(3) for a description of how to initialize the environment and how to start a Perl program):

```
[ /home/fred ]
[ fred@acme-007 ][bash]: eval `~edrc/lib/perlenv`
```

Perl program using **WA2L::Util**:

```
#!/usr/bin/env perl

use WA2L::Util;
use File::Basename;

$ENV{EDRC_LOGFILE} = approot() . "/var/log/" .
    basename($0) . ".log";

# print message
#
msg( level => "INFO", txt => "start script" );

# create a log file entry
#
log( level => "INFO", txt => "start script" );

# print timestamp
#
print now() . "\n";

# check options
#
checkopt(
    used_opts=>"fv",
    truthtable => "
        fnv
        100
        101
        110
    "
) or die('Usage: ' . basename $0 . ' -f file [ -n | -v ]');
```

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is a collection of general WA2L/edrc utility functions to be used in Perl scripts or other Perl modules.

FUNCTIONS

msg(level => "LEVEL", txt => "text")

print a standardized message text to **stderr**.

See **msg(3)** for the message format.

log(level => "LEVEL", txt => "text")

write a standardized message text to the log file specified in the **\$EDRC_LOGFILE** environment variable. The session can be set in the **\$EDRC_SESSION**. If **\$EDRC_SESSION** is not set, the process ID of the calling Perl script is used as a ad-hoc session id.

See **log(3)** for the output format.

now()

return the current date and time in the military format **YYYY-MM-DD hh:mm:ss**.

approot()

return the root of the WA2L/edrc package installation. The output is identical to the **approot(3)** command.

checkopt(used_opts => "options", truthtable => "table")

check if a list of used *options* matches against a *table* of allowed option combinations. This command is intended to be used to efficiently parse arguments in a script.

The **checkopt** function returns **1** (=true) if *options* are correct, **0** (=false) if *options* are not correct.

See **checkopt(3)** for the detail description.

ENVIRONMENT

\$EDRC_SESSION

session name.

\$EDRC_LOGFILE

log file name and path.

EXIT STATUS

-

FILES

edrc/lib/perl/pm/perl*/vendor_perl/WA2L/Util.pm

this Perl module file.

EXAMPLES

-

SEE ALSO

edrcintro(1), **perl(1)**, **perlenv(3)**, **perl_modules(3)**

NOTES

-

BUGS

-

AUTHOR

WA2L::Util was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2021 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

watchdog – a watchdog to check a condition and react on it

SYNOPSIS

edrc/bin/watchdog [**-h**]

watchdog **-n** *name* **-c** *check_script* **-b** *bite_script* [**-i** *c_int* [, *b_int*]] [**-r** *l_int*]

watchdog **-s** *name*

watchdog **-l** | **-L** { [**start**] | **stop** }

AVAILABILITY

WA2L/edrc

DESCRIPTION

The intention of the **watchdog** command is to easily monitor a certain condition or service and react if the condition is not OK.

The states and terminology of the watchdog is adapted from the behavior of a real dog which checks, waits (between checks), bites (if something is not OK) and breaths (to recover between bites).

The different possible states of all running watchdogs are logged to a logfile.

The check script has to be written that way that a successful check returns an exit code of 0. An unsuccessful check has to return an exit code $\neq 0$. The interval between check script starts (= check interval) can be specified with the **-i** option.

The bite script (= reaction to an unsuccessful check) has to be written that way that a successful bite returns an exit code of 0. An unsuccessful bite should return an exit code $\neq 0$ to ensure a direct bite script restart. The interval between bite script restarts (= bite interval) can be specified with the **-i** option.

The bite script and the check script must exist on startup of a watchdog.

If a bite- or check script disappears during operation the number of retries can be configured in the configuration file **watchdog.cfg** with the **RETRIES_ON_MISSING_SCRIPT** setting.

The watchdog has the following states which are also recorded to the logfile:

check This is also the first state after the start of a watchdog. Execute the check script. If the check script returns an exit code of 0 goto state "wait", else goto state "bite".

To avoid logfile fill ups of subsequent "check" state recordings, the "check" state is logged in a bigger interval, which can be specified with the **-r** option if there is a need to deviate from the default of an hour.

| | |
|---------------|--|
| bite | Execute the bite script which hopefully can restart the service or recreate the proper condition. If the bite script returns an exit code of 0 goto state "check", else goto state "breath". |
| wait | Wait the duration of seconds specified as the <i>check_interval</i> , then goto state "check". |
| breath | Wait the duration of seconds specified as the <i>bite_interval</i> , then goto state "bite". |
| stop | The watchdog has been stopped with the -s option. |
| abort | The watchdog has been killed with the kill watchdog_process_id command (not recommended). |

OPTIONS

| | |
|------------------------|--|
| -h | usage message. |
| -n name | name of the watchdog. |
| -c check_script | script or single command that performs a check. If this script returns an exit code $\neq 0$ then the bite script specified with the -b option is started. |
| -b bite_script | script or single command that performs the reaction (bite) if the check script specified with the -c option returned an exit code $\neq 0$. |
| -i c_int | [, <i>b_int</i>] <i>c_int</i> check interval in seconds. Every <i>c_int</i> seconds the script specified in option -c is executed. If -i is not specified the default check interval of 500 seconds applies. <i>b_int</i> bite interval in seconds. This is the interval between the execution of the <i>bite_script</i> if the bite script returns an exit code $\neq 0$, that means between unsuccessful reactions (bites) of the script specified in option -b . If the <i>b_int</i> is not specified the default bite interval of 20 seconds applies. |
| -r l_int | minimal interval in seconds between logging of the check state message. If this option is not specified the default of 3600 seconds applies. |
| -s name | Stop a running watchdog with the name <i>name</i> . Users can only stop own started watchdogs. Be aware that a stop of a watchdog can be delayed by the duration of the check- or bite script run. You should not kill the watchdog with the kill command. |
| -l | list all running watchdogs. |

-L { [start] | stop }

list start/stop commands for all running watchdogs. This enables you to easily restart all currently running watchdogs with the related command options.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 1** the configuration file **edrc/etc/watchdog.cfg** does not exist.
- 2** operating system not supported, see **osid(3)**
- 3** cannot write to lock directory. The ability for all user to write to the lockdir is mandatory.
- 4** usage displayed.
- 5** a watchdog with the same name as specified in the **-n** option is already running.
- 6** the specified check script does not exist during watchdog startup.
- 7** the specified bite script does not exist during watchdog startup.
- 8** cannot write to logfile. The ability for all users to write to the logfile is mandatory.
- 9** unsuccessful attempt to stop a watchdog of an other user.
- 10** no watchdog is running while listing the started watchdogs with the **-l** or **-L** option.
- 11** temporary directory could not be claimed or created in **/var/tmp**. Check the system temporary directory **/var/tmp** if you get this error, it is an indicator of system intrusion.

FILES

edrc/etc/watchdog.cfg

configuration file of **watchdog**. See **watchdog.cfg(4)** for more information.

edrc/var/watchdog

default lockdir. This directory holds the lockfiles of watchdog. Do not edit them by hand. The lockdir can be configured in **watchdog.cfg**, see **watchdog.cfg(4)** for more information.

EXAMPLES

1) using watchdog to guard the AutoSYS event daemon

The AutoSYS event daemon is a program that is connected to a database and initiates job starts on remote systems. In some versions of AutoSYS the daemon tends to shut down after a startup. To ensure the permanent availability of the AutoSYS job scheduler the event daemon is guarded by a watchdog.

1.1) check script

The check script uses the AutoSYS command "chk_auto_up" to verify if the event daemon is up and running and returns 0 if this is true.

```
#!/bin/ksh
#
# eventor.check - check the event_daemon is running
#
# [00] 13.08.2004 CWa Initial Version
#
/bin/su - sys_asys -c "chk_auto_up"
if [ $? -eq 11 ]; then
    exit 0
else
    exit 1
fi
```

1.2) bite script

The bite script uses the AutoSYS command "eventor -q" to start the event daemon and the command "chk_auto_up" to verify if the start of the event daemon was successful. If the start was successful the script returns 0.

```
#!/bin/ksh
#
# eventor.bite - restart event_daemon if it not available
#
# [00] 13.08.2004 CWa Initial Version
#
/bin/su - sys_asys -c "eventor -q"
sleep 5

/bin/su - sys_asys -c "chk_auto_up"
if [ $? -eq 11 ]; then
    exit 0
else
    exit 1
fi
```

1.3) watchdog startup

This watchdog checks every 5 minutes if the event daemon is up and bites every 30 seconds if the event daemon start was not successful. Check states are logged to the logfile once approximately every 30 minutes.

```
watchdog -n event_daemon \  
-c /etc/cmcluster/asys_sv1_prod/eventor.check \  
-b /etc/cmcluster/asys_sv1_prod/eventor.bite \  
-i 300,30 \  
-r 1800
```

SEE ALSO

edrcintro(1), **osid(3)**, **sh(1)**, **watchdog.cfg(4)**

NOTES

-

BUGS

-

AUTHOR

watchdog was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

watchdog.cfg – configuration file for watchdog

SYNOPSIS

edrc/etc/watchdog.cfg

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the configuration file for the **watchdog** command.

FILEFORMAT

The fileformat is **OPTION=VALUE**

Between the **OPTION**, the = and the *VALUE* are no spaces.

Rows starting with a # are considered as comments.

You should not comment out any **OPTION** If you like to use default settings simply do not specify a *VALUE*.

OPTIONS

LOG Log output dir of **watchdog**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation.

Example: LOG=/var/opt/edrc/log

Default: LOG=var/log

LOCKDIR

Lock dir of **watchdog**. If you specify a relative path name the path is relative to the root of the WA2L/edrc installation. In General it is not recommended to set the lockdir within EDRC, locate it in a system own directory.

Example: LOCKDIR=/var/run/watchdog

Default: LOCKDIR=var/watchdog

DEFAULT_CHECK_INTERVAL

Default interval of watchdog checks. This setting applies if no interval is set on the command line.

Example: DEFAULT_CHECK_INTERVAL=500

Default: DEFAULT_CHECK_INTERVAL=600

DEFAULT_CHECKLOG_INTERVAL

Default minimal interval of logentries in check state. This setting applies if no interval is set on the command line. This setting should be an even multiple of DEFAULT_CHECK_INTERVAL.

Example: DEFAULT_CHECKLOG_INTERVAL=3600

Default: DEFAULT_CHECKLOG_INTERVAL=6000

DEFAULT_BITE_INTERVAL

Default interval of **watchdog** bites (=executes of the bite script). This setting applies if no interval is set on the command line.

Example: DEFAULT_BITE_INTERVAL=15

Default: DEFAULT_BITE_INTERVAL=5

EXEC_SHELL

Shell to execute the check_script and bite_script if they are not executable.

Example: EXEC_SHELL=/bin/ksh

Default: EXEC_SHELL=/bin/sh

RETRIES_ON_MISSING_SCRIPT

Number of retries if the check_script or bite_script is missing after startup. If the number of retries is reached the **watchdog** will end.

0 = retry for ever

Example: RETRIES_ON_MISSING_SCRIPT=10

Default: RETRIES_ON_MISSING_SCRIPT=0

SEE ALSO

edrcintro(1), **watchdog(1)**, **sh(1)**

NOTES

-

BUGS

-

AUTHOR

watchdog.cfg was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

Wget – The non-interactive network downloader.

SYNOPSIS

```
wget [option]... [URL]...
```

DESCRIPTION

GNU Wget is a free utility for non-interactive download of files from the Web. It supports HTTP, HTTPS, and FTP protocols, as well as retrieval through HTTP proxies.

Wget is non-interactive, meaning that it can work in the background, while the user is not logged on. This allows you to start a retrieval and disconnect from the system, letting Wget finish the work. By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data.

Wget can follow links in HTML and XHTML pages and create local versions of remote web sites, fully recreating the directory structure of the original site. This is sometimes referred to as “recursive downloading.” While doing that, Wget respects the Robot Exclusion Standard (*/robots.txt*). Wget can be instructed to convert the links in downloaded HTML files to the local files for offline viewing.

Wget has been designed for robustness over slow or unstable network connections; if a download fails due to a network problem, it will keep retrying until the whole file has been retrieved. If the server supports regetting, it will instruct the server to continue the download from where it left off.

OPTIONS**Option Syntax**

Since Wget uses GNU getopt to process command-line arguments, every option has a long form along with the short one. Long options are more convenient to remember, but take time to type. You may freely mix different option styles, or specify options after the command-line arguments. Thus you may write:

```
wget -r --tries=10 http://fly.srk.fer.hr/ -o log
```

The space between the option accepting an argument and the argument may be omitted. Instead **-o log** you can write **-olog**.

You may put several options that do not require arguments together, like:

```
wget -drc <URL>
```

This is a complete equivalent of:

```
wget -d -r -c <URL>
```

Since the options can be specified after the arguments, you may terminate them with **--**. So the following will try to download URL **-x**, reporting failure to *log*:

```
wget -o log -- -x
```

The options that accept comma-separated lists all respect the convention that specifying an empty list clears its value. This can be useful to clear the *.wgetrc* settings. For instance, if your *.wgetrc* sets *exclude_directories* to */cgi-bin*, the following example will first reset it, and then set it to exclude */nobody* and */somebody*. You can also clear the lists in *.wgetrc*.

```
wget -X '' -X /~nobody,/~somebody
```

Most options that do not accept arguments are *boolean* options, so named because their state can be captured with a yes-or-no (“boolean”) variable. For example, **—follow-ftp** tells Wget to follow FTP links from HTML files and, on the other hand, **—no-glob** tells it not to perform file globbing on FTP URLs. A boolean option is either *affirmative* or *negative* (beginning with **—no**). All such options share several properties.

Unless stated otherwise, it is assumed that the default behavior is the opposite of what the option

accomplishes. For example, the documented existence of **—follow-ftp** assumes that the default is to *not* follow FTP links from HTML pages.

Affirmative options can be negated by prepending the **—no-** to the option name; negative options can be negated by omitting the **—no-** prefix. This might seem superfluous---if the default for an affirmative option is to not do something, then why provide a way to explicitly turn it off? But the startup file may in fact change the default. For instance, using `follow_ftp = off` in `.wgetrc` makes Wget *not* follow FTP links by default, and using **—no-follow-ftp** is the only way to restore the factory default from the command line.

Basic Startup Options

-V

—version

Display the version of Wget.

-h

—help

Print a help message describing all of Wget's command-line options.

-b

—background

Go to background immediately after startup. If no output file is specified via the **-o**, output is redirected to `wget-log`.

-e command

—execute command

Execute *command* as if it were a part of `.wgetrc`. A command thus invoked will be executed *after* the commands in `.wgetrc`, thus taking precedence over them. If you need to specify more than one wgetrc command, use multiple instances of **-e**.

Logging and Input File Options

-o logfile

—output-file=logfile

Log all messages to *logfile*. The messages are normally reported to standard error.

-a logfile

—append-output=logfile

Append to *logfile*. This is the same as **-o**, only it appends to *logfile* instead of overwriting the old log file. If *logfile* does not exist, a new file is created.

-d

—debug

Turn on debug output, meaning various information important to the developers of Wget if it does not work properly. Your system administrator may have chosen to compile Wget without debug support, in which case **-d** will not work. Please note that compiling with debug support is always safe---Wget compiled with the debug support will *not* print any debug info unless requested with **-d**.

-q

—quiet

Turn off Wget's output.

-v

—verbose

Turn on verbose output, with all the available data. The default output is verbose.

-nv

—no-verbose

Turn off verbose without being completely quiet (use **-q** for that), which means that error messages and basic information still get printed.

-i file

—input-file=file

Read URLs from *file*. If *-* is specified as *file*, URLs are read from the standard input. (Use **-J-** to read from a file literally named *-*.)

If this function is used, no URLs need be present on the command line. If there are URLs both on the command line and in an input file, those on the command lines will be the first ones to be retrieved. The *file* need not be an HTML document (but no harm if it is)---it is enough if the URLs are just listed sequentially.

However, if you specify **—force-html**, the document will be regarded as **html**. In that case you may have problems with relative links, which you can solve either by adding `<base href="url">` to HTML, or to the documents or by specifying **—base=url** on the command line.

-F

—force-html

When input is read from a file, force it to be treated as an HTML file. This enables you to retrieve relative links from existing HTML files on your local disk, by adding `<base href="url">` to HTML, or using the **—base** command-line option.

-B URL

—base=URL

Prepends *URL* to relative links read from the file specified with the **-i** option.

Download Options

—bind-address=ADDRESS

When making client TCP/IP connections, bind to *ADDRESS* on the local machine. *ADDRESS* may be specified as a hostname or IP address. This option can be useful if your machine is bound to multiple IPs.

-t number

—tries=number

Set number of retries to *number*. Specify 0 or **inf** for infinite retrying. The default is to retry 20 times, with the exception of fatal errors like “connection refused” or “not found” (404), which are not retried.

-O file

—output-document=file

The documents will not be written to the appropriate files, but all will be concatenated together and written to *file*. If *-* is used as *file*, documents will be printed to standard output, disabling link conversion. (Use **-J-** to print to a file literally named *-*.)

Note that a combination with **-k** is only well-defined for downloading a single document.

-nc

—no-clobber

If a file is downloaded more than once in the same directory, Wget’s behavior depends on a few options, including **-nc**. In certain cases, the local file will be *clobbered*, or overwritten, upon repeated download. In other cases it will be preserved.

When running Wget without **-N**, **-nc**, or **-r**, downloading the same file in the same directory will result in the original copy of *file* being preserved and the second copy being named *file.1*. If that file is

downloaded yet again, the third copy will be named *file.2*, and so on. When **-nc** is specified, this behavior is suppressed, and Wget will refuse to download newer copies of *file*. Therefore, “no-clobber” is actually a misnomer in this mode—it’s not clobbering that’s prevented (as the numeric suffixes were already preventing clobbering), but rather the multiple version saving that’s prevented.

When running Wget with **-r**, but without **-N** or **-nc**, re-downloading a file will result in the new copy simply overwriting the old. Adding **-nc** will prevent this behavior, instead causing the original version to be preserved and any newer copies on the server to be ignored.

When running Wget with **-N**, with or without **-r**, the decision as to whether or not to download a newer copy of a file depends on the local and remote timestamp and size of the file. **-nc** may not be specified at the same time as **-N**.

Note that when **-nc** is specified, files with the suffixes **.html** or **.htm** will be loaded from the local disk and parsed as if they had been retrieved from the Web.

-c

—continue

Continue getting a partially-downloaded file. This is useful when you want to finish up a download started by a previous instance of Wget, or by another program. For instance:

```
wget -c ftp://sunsite.doc.ic.ac.uk/ls-lR.Z
```

If there is a file named *ls-lR.Z* in the current directory, Wget will assume that it is the first portion of the remote file, and will ask the server to continue the retrieval from an offset equal to the length of the local file.

Note that you don’t need to specify this option if you just want the current invocation of Wget to retry downloading a file should the connection be lost midway through. This is the default behavior. **-c** only affects resumption of downloads started *prior* to this invocation of Wget, and whose local files are still sitting around.

Without **-c**, the previous example would just download the remote file to *ls-lR.Z.1*, leaving the truncated *ls-lR.Z* file alone.

Beginning with Wget 1.7, if you use **-c** on a non-empty file, and it turns out that the server does not support continued downloading, Wget will refuse to start the download from scratch, which would effectively ruin existing contents. If you really want the download to start from scratch, remove the file.

Also beginning with Wget 1.7, if you use **-c** on a file which is of equal size as the one on the server, Wget will refuse to download the file and print an explanatory message. The same happens when the file is smaller on the server than locally (presumably because it was changed on the server since your last download attempt)—because “continuing” is not meaningful, no download occurs.

On the other side of the coin, while using **-c**, any file that’s bigger on the server than locally will be considered an incomplete download and only $(\text{length}(\text{remote}) - \text{length}(\text{local}))$ bytes will be downloaded and tacked onto the end of the local file. This behavior can be desirable in certain cases—for instance, you can use **wget -c** to download just the new portion that’s been appended to a data collection or log file.

However, if the file is bigger on the server because it’s been *changed*, as opposed to just *appended* to, you’ll end up with a garbled file. Wget has no way of verifying that the local file is really a valid prefix of the remote file. You need to be especially careful of this when using **-c** in conjunction with **-r**, since every file will be considered as an “incomplete download” candidate.

Another instance where you’ll get a garbled file if you try to use **-c** is if you have a lame HTTP proxy that inserts a “transfer interrupted” string into the local file. In the future a “rollback” option may be added to deal with this case.

Note that **-c** only works with FTP servers and with HTTP servers that support the Range header.

—progress=*type*

Select the type of the progress indicator you wish to use. Legal indicators are “dot” and “bar”.

The “bar” indicator is used by default. It draws an ASCII progress bar graphics (a.k.a “thermometer” display) indicating the status of retrieval. If the output is not a TTY, the “dot” bar will be used by default.

Use **—progress=dot** to switch to the “dot” display. It traces the retrieval by printing dots on the screen, each dot representing a fixed amount of downloaded data.

When using the dotted retrieval, you may also set the *style* by specifying the type as **dot:style**. Different styles assign different meaning to one dot. With the default style each dot represents 1K, there are ten dots in a cluster and 50 dots in a line. The binary style has a more “computer”-like orientation—8K dots, 16-dots clusters and 48 dots per line (which makes for 384K lines). The mega style is suitable for downloading very large files—each dot represents 64K retrieved, there are eight dots in a cluster, and 48 dots on each line (so each line contains 3M).

Note that you can set the default style using the `progress` command in `.wgetrc`. That setting may be overridden from the command line. The exception is that, when the output is not a TTY, the “dot” progress will be favored over “bar”. To force the bar output, use **—progress=bar:force**.

-N

—timestamping

Turn on time-stamping.

-S

—server-response

Print the headers sent by HTTP servers and responses sent by FTP servers.

—spider

When invoked with this option, Wget will behave as a Web *spider*, which means that it will not download the pages, just check that they are there. For example, you can use Wget to check your bookmarks:

```
wget --spider --force-html -i bookmarks.html
```

This feature needs much more work for Wget to get close to the functionality of real web spiders.

-T seconds

—timeout=*seconds*

Set the network timeout to *seconds* seconds. This is equivalent to specifying **—dns-timeout**, **—connect-timeout**, and **—read-timeout**, all at the same time.

When interacting with the network, Wget can check for timeout and abort the operation if it takes too long. This prevents anomalies like hanging reads and infinite connects. The only timeout enabled by default is a 900-second read timeout. Setting a timeout to 0 disables it altogether. Unless you know what you are doing, it is best not to change the default timeout settings.

All timeout-related options accept decimal values, as well as subsecond values. For example, **0.1** seconds is a legal (though unwise) choice of timeout. Subsecond timeouts are useful for checking server response times or for testing network latency.

—dns-timeout=*seconds*

Set the DNS lookup timeout to *seconds* seconds. DNS lookups that don’t complete within the specified time will fail. By default, there is no timeout on DNS lookups, other than that implemented by system libraries.

—connect-timeout=seconds

Set the connect timeout to *seconds* seconds. TCP connections that take longer to establish will be aborted. By default, there is no connect timeout, other than that implemented by system libraries.

—read-timeout=seconds

Set the read (and write) timeout to *seconds* seconds. The “time” of this timeout refers *idle time*: if, at any point in the download, no data is received for more than the specified number of seconds, reading fails and the download is restarted. This option does not directly affect the duration of the entire download.

Of course, the remote server may choose to terminate the connection sooner than this option requires. The default read timeout is 900 seconds.

—limit-rate=amount

Limit the download speed to *amount* bytes per second. Amount may be expressed in bytes, kilobytes with the **k** suffix, or megabytes with the **m** suffix. For example, **—limit-rate=20k** will limit the retrieval rate to 20KB/s. This is useful when, for whatever reason, you don’t want Wget to consume the entire available bandwidth.

This option allows the use of decimal numbers, usually in conjunction with power suffixes; for example, **—limit-rate=2.5k** is a legal value.

Note that Wget implements the limiting by sleeping the appropriate amount of time after a network read that took less time than specified by the rate. Eventually this strategy causes the TCP transfer to slow down to approximately the specified rate. However, it may take some time for this balance to be achieved, so don’t be surprised if limiting the rate doesn’t work well with very small files.

—w seconds**—wait=seconds**

Wait the specified number of seconds between the retrievals. Use of this option is recommended, as it lightens the server load by making the requests less frequent. Instead of in seconds, the time can be specified in minutes using the **m** suffix, in hours using **h** suffix, or in days using **d** suffix.

Specifying a large value for this option is useful if the network or the destination host is down, so that Wget can wait long enough to reasonably expect the network error to be fixed before the retry.

—waitretry=seconds

If you don’t want Wget to wait between *every* retrieval, but only between retries of failed downloads, you can use this option. Wget will use *linear backoff*, waiting 1 second after the first failure on a given file, then waiting 2 seconds after the second failure on that file, up to the maximum number of *seconds* you specify. Therefore, a value of 10 will actually make Wget wait up to $(1 + 2 + \dots + 10) = 55$ seconds per file.

Note that this option is turned on by default in the global *wgetrc* file.

—random-wait

Some web sites may perform log analysis to identify retrieval programs such as Wget by looking for statistically significant similarities in the time between requests. This option causes the time between requests to vary between 0 and $2 * \text{wait}$ seconds, where *wait* was specified using the **—wait** option, in order to mask Wget’s presence from such analysis.

A recent article in a publication devoted to development on a popular consumer platform provided code to perform this analysis on the fly. Its author suggested blocking at the class C address level to ensure automated retrieval programs were blocked despite changing DHCP-supplied addresses.

The **—random-wait** option was inspired by this ill-advised recommendation to block many unrelated users from a web site due to the actions of one.

—no-proxy

Don’t use proxies, even if the appropriate **_proxy* environment variable is defined.

For more information about the use of proxies with Wget,

-Q quota**—quota=quota**

Specify download quota for automatic retrievals. The value can be specified in bytes (default), kilobytes (with **k** suffix), or megabytes (with **m** suffix).

Note that quota will never affect downloading a single file. So if you specify **wget -Q10k ftp://wuarchive.wustl.edu/ls-lR.gz**, all of the *ls-lR.gz* will be downloaded. The same goes even when several URLs are specified on the command-line. However, quota is respected when retrieving either recursively, or from an input file. Thus you may safely type **wget -Q2m -i sites**—download will be aborted when the quota is exceeded.

Setting quota to 0 or to **inf** unlimits the download quota.

—no-dns-cache

Turn off caching of DNS lookups. Normally, Wget remembers the IP addresses it looked up from DNS so it doesn't have to repeatedly contact the DNS server for the same (typically small) set of hosts it retrieves from. This cache exists in memory only; a new Wget run will contact DNS again.

However, it has been reported that in some situations it is not desirable to cache host names, even for the duration of a short-running application like Wget. With this option Wget issues a new DNS lookup (more precisely, a new call to `gethostbyname` or `getaddrinfo`) each time it makes a new connection. Please note that this option will *not* affect caching that might be performed by the resolving library or by an external caching layer, such as NSCD.

If you don't understand exactly what this option does, you probably won't need it.

—restrict-file-names=mode

Change which characters found in remote URLs may show up in local file names generated from those URLs. Characters that are *restricted* by this option are escaped, i.e. replaced with **%HH**, where **HH** is the hexadecimal number that corresponds to the restricted character.

By default, Wget escapes the characters that are not valid as part of file names on your operating system, as well as control characters that are typically unprintable. This option is useful for changing these defaults, either because you are downloading to a non-native partition, or because you want to disable escaping of the control characters.

When mode is set to "unix", Wget escapes the character **/** and the control characters in the ranges 0—31 and 128—159. This is the default on Unix-like OS'es.

When mode is set to "windows", Wget escapes the characters ****, **|**, **/**, **:**, **?**, **"**, *****, **<**, **>**, and the control characters in the ranges 0—31 and 128—159. In addition to this, Wget in Windows mode uses **+** instead of **:** to separate host and port in local file names, and uses **@** instead of **?** to separate the query portion of the file name from the rest. Therefore, a URL that would be saved as **www.xemacs.org:4300/search.pl?input=blah** in Unix mode would be saved as **www.xemacs.org+4300/search.pl@input=blah** in Windows mode. This mode is the default on Windows.

If you append **,nocontrol** to the mode, as in **unix,nocontrol**, escaping of the control characters is also switched off. You can use **—restrict-file-names=nocontrol** to turn off escaping of control characters without affecting the choice of the OS to use as file name restriction mode.

-4**—inet4-only****-6****—inet6-only**

Force connecting to IPv4 or IPv6 addresses. With **—inet4-only** or **-4**, Wget will only connect to IPv4 hosts, ignoring AAAA records in DNS, and refusing to connect to IPv6 addresses specified in URLs. Conversely, with **—inet6-only** or **-6**, Wget will only connect to IPv6 hosts and ignore A records and IPv4 addresses.

Neither options should be needed normally. By default, an IPv6-aware Wget will use the address family specified by the host's DNS record. If the DNS responds with both IPv4 and IPv6 addresses, Wget will try them in sequence until it finds one it can connect to. (Also see `--prefer-family` option described below.)

These options can be used to deliberately force the use of IPv4 or IPv6 address families on dual family systems, usually to aid debugging or to deal with broken network configuration. Only one of **—inet6-only** and **—inet4-only** may be specified at the same time. Neither option is available in Wget compiled without IPv6 support.

—prefer-family=IPv4/IPv6/none

When given a choice of several addresses, connect to the addresses with specified address family first. IPv4 addresses are preferred by default.

This avoids spurious errors and connect attempts when accessing hosts that resolve to both IPv6 and IPv4 addresses from IPv4 networks. For example, **www.kame.net** resolves to **2001:200:0:8002:203:47ff:fea5:3085** and to **203.178.141.194**. When the preferred family is IPv4, the IPv4 address is used first; when the preferred family is IPv6, the IPv6 address is used first; if the specified value is `none`, the address order returned by DNS is used without change.

Unlike **-4** and **-6**, this option doesn't inhibit access to any address family, it only changes the *order* in which the addresses are accessed. Also note that the reordering performed by this option is *stable*—it doesn't affect order of addresses of the same family. That is, the relative order of all IPv4 addresses and of all IPv6 addresses remains intact in all cases.

—retry-connrefused

Consider “connection refused” a transient error and try again. Normally Wget gives up on a URL when it is unable to connect to the site because failure to connect is taken as a sign that the server is not running at all and that retries would not help. This option is for mirroring unreliable sites whose servers tend to disappear for short periods of time.

—user=*user*

—password=*password*

Specify the username *user* and password *password* for both FTP and HTTP file retrieval. These parameters can be overridden using the **—ftp-user** and **—ftp-password** options for FTP connections and the **—http-user** and **—http-password** options for HTTP connections.

Directory Options

—nd

—no-directories

Do not create a hierarchy of directories when retrieving recursively. With this option turned on, all files will get saved to the current directory, without clobbering (if a name shows up more than once, the filenames will get extensions **.n**).

—x

—force-directories

The opposite of **—nd**—create a hierarchy of directories, even if one would not have been created otherwise. E.g. **wget -x http://fly.srk.fer.hr/robots.txt** will save the downloaded file to *fly.srk.fer.hr/robots.txt*.

—nH

—no-host-directories

Disable generation of host-prefixed directories. By default, invoking Wget with **—r http://fly.srk.fer.hr/** will create a structure of directories beginning with *fly.srk.fer.hr/*. This option disables such behavior.

—protocol-directories

Use the protocol name as a directory component of local file names. For example, with this option, **wget -r http://host** will save to **http/host/...** rather than just to **host/...**

—cut-dirs=number

Ignore *number* directory components. This is useful for getting a fine-grained control over the directory where recursive retrieval will be saved.

Take, for example, the directory at **ftp://ftp.xemacs.org/pub/xemacs/**. If you retrieve it with **-r**, it will be saved locally under **ftp.xemacs.org/pub/xemacs/**. While the **-nH** option can remove the **ftp.xemacs.org/** part, you are still stuck with **pub/xemacs**. This is where **—cut-dirs** comes in handy; it makes Wget not “see” *number* remote directory components. Here are several examples of how **—cut-dirs** option works.

```

No options          -> ftp.xemacs.org/pub/xemacs/
-nH                 -> pub/xemacs/
-nH --cut-dirs=1    -> xemacs/
-nH --cut-dirs=2    -> .

--cut-dirs=1        -> ftp.xemacs.org/xemacs/
...

```

If you just want to get rid of the directory structure, this option is similar to a combination of **-nd** and **-P**. However, unlike **-nd**, **—cut-dirs** does not lose with subdirectories—for instance, with **-nH --cut-dirs=1**, a *beta/* subdirectory will be placed to *xemacs/beta*, as one would expect.

-P prefix**—directory-prefix=prefix**

Set directory prefix to *prefix*. The *directory prefix* is the directory where all other files and subdirectories will be saved to, i.e. the top of the retrieval tree. The default is **.** (the current directory).

HTTP Options**-E****—html-extension**

If a file of type **application/xhtml+xml** or **text/html** is downloaded and the URL does not end with the regexp **\[Hh][Tt][Mm][Ll]?**, this option will cause the suffix **.html** to be appended to the local filename. This is useful, for instance, when you’re mirroring a remote site that uses **.asp** pages, but you want the mirrored pages to be viewable on your stock Apache server. Another good use for this is when you’re downloading CGI-generated materials. A URL like **http://site.com/article.cgi?25** will be saved as *article.cgi?25.html*.

Note that filenames changed in this way will be re-downloaded every time you re-mirror a site, because Wget can’t tell that the local *X.html* file corresponds to remote URL *X* (since it doesn’t yet know that the URL produces output of type **text/html** or **application/xhtml+xml**). To prevent this re-downloading, you must use **-k** and **-K** so that the original version of the file will be saved as *X.orig*.

—http-user=user**—http-password=password**

Specify the username *user* and password *password* on an HTTP server. According to the type of the challenge, Wget will encode them using either the **basic** (insecure) or the **digest** authentication scheme.

Another way to specify username and password is in the URL itself. Either method reveals your password to anyone who bothers to run **ps**. To prevent the passwords from being seen, store them in *.wgetrc* or *.netrc*, and make sure to protect those files from other users with **chmod**. If the passwords are really important, do not leave them lying in those files either—edit the files and delete them after Wget has started the download.

—no-cache

Disable server-side cache. In this case, Wget will send the remote server an appropriate directive (**Pragma: no-cache**) to get the file from the remote service, rather than returning the cached version. This is especially useful for retrieving and flushing out-of-date documents on proxy servers.

Caching is allowed by default.

—no-cookies

Disable the use of cookies. Cookies are a mechanism for maintaining server-side state. The server sends the client a cookie using the `Set-Cookie` header, and the client responds with the same cookie upon further requests. Since cookies allow the server owners to keep track of visitors and for sites to exchange this information, some consider them a breach of privacy. The default is to use cookies; however, *storing* cookies is not on by default.

—load-cookies file

Load cookies from *file* before the first HTTP retrieval. *file* is a textual file in the format originally used by Netscape's *cookies.txt* file.

You will typically use this option when mirroring sites that require that you be logged in to access some or all of their content. The login process typically works by the web server issuing an HTTP cookie upon receiving and verifying your credentials. The cookie is then resent by the browser when accessing that part of the site, and so proves your identity.

Mirroring such a site requires Wget to send the same cookies your browser sends when communicating with the site. This is achieved by **—load-cookies**—simply point Wget to the location of the *cookies.txt* file, and it will send the same cookies your browser would send in the same situation. Different browsers keep textual cookie files in different locations:

Netscape 4.x.

The cookies are in *~/netscape/cookies.txt*.

Mozilla and Netscape 6.x.

Mozilla's cookie file is also named *cookies.txt*, located somewhere under *~/mozilla*, in the directory of your profile. The full path usually ends up looking somewhat like *~/mozilla/default/some-weird-string/cookies.txt*.

Internet Explorer.

You can produce a cookie file Wget can use by using the File menu, Import and Export, Export Cookies. This has been tested with Internet Explorer 5; it is not guaranteed to work with earlier versions.

Other browsers.

If you are using a different browser to create your cookies, **—load-cookies** will only work if you can locate or produce a cookie file in the Netscape format that Wget expects.

If you cannot use **—load-cookies**, there might still be an alternative. If your browser supports a “cookie manager”, you can use it to view the cookies used when accessing the site you're mirroring. Write down the name and value of the cookie, and manually instruct Wget to send those cookies, bypassing the “official” cookie support:

```
wget --no-cookies --header "Cookie: <name>=<value>"
```

—save-cookies file

Save cookies to *file* before exiting. This will not save cookies that have expired or that have no expiry time (so-called “session cookies”), but also see **—keep-session-cookies**.

—keep-session-cookies

When specified, causes **—save-cookies** to also save session cookies. Session cookies are normally not saved because they are meant to be kept in memory and forgotten when you exit the browser. Saving them is useful on sites that require you to log in or to visit the home page before you can access some pages. With this option, multiple Wget runs are considered a single browser session as far as the

site is concerned.

Since the cookie file format does not normally carry session cookies, Wget marks them with an expiry timestamp of 0. Wget's **—load-cookies** recognizes those as session cookies, but it might confuse other browsers. Also note that cookies so loaded will be treated as other session cookies, which means that if you want **—save-cookies** to preserve them again, you must use **—keep-session-cookies** again.

—ignore-length

Unfortunately, some HTTP servers (CGI programs, to be more precise) send out bogus `Content-Length` headers, which makes Wget go wild, as it thinks not all the document was retrieved. You can spot this syndrome if Wget retries getting the same document again and again, each time claiming that the (otherwise normal) connection has closed on the very same byte.

With this option, Wget will ignore the `Content-Length` header—as if it never existed.

—header=header-line

Send *header-line* along with the rest of the headers in each HTTP request. The supplied header is sent as-is, which means it must contain name and value separated by colon, and must not contain newlines.

You may define more than one additional header by specifying **—header** more than once.

```
wget --header='Accept-Charset: iso-8859-2' \
      --header='Accept-Language: hr' \
      http://fly.srk.fer.hr/
```

Specification of an empty string as the header value will clear all previous user-defined headers.

As of Wget 1.10, this option can be used to override headers otherwise generated automatically. This example instructs Wget to connect to localhost, but to specify **foo.bar** in the `Host` header:

```
wget --header="Host: foo.bar" http://localhost/
```

In versions of Wget prior to 1.10 such use of **—header** caused sending of duplicate headers.

—proxy-user=user

—proxy-password=password

Specify the username *user* and password *password* for authentication on a proxy server. Wget will encode them using the `basic` authentication scheme.

Security considerations similar to those with **—http-password** pertain here as well.

—referer=url

Include 'Referer: *url*' header in HTTP request. Useful for retrieving documents with server-side processing that assume they are always being retrieved by interactive web browsers and only come out properly when Referer is set to one of the pages that point to them.

—save-headers

Save the headers sent by the HTTP server to the file, preceding the actual contents, with an empty line as the separator.

—U agent-string

—user-agent=agent-string

Identify as *agent-string* to the HTTP server.

The HTTP protocol allows the clients to identify themselves using a `User-Agent` header field. This enables distinguishing the WWW software, usually for statistical purposes or for tracing of protocol violations. Wget normally identifies as **Wget/version**, *version* being the current version number of Wget.

However, some sites have been known to impose the policy of tailoring the output according to the `User-Agent`-supplied information. While this is not such a bad idea in theory, it has been abused by servers denying information to clients other than (historically) Netscape or, more frequently,

Microsoft Internet Explorer. This option allows you to change the `User-Agent` line issued by Wget. Use of this option is discouraged, unless you really know what you are doing.

Specifying empty user agent with `—user-agent=""` instructs Wget not to send the `User-Agent` header in HTTP requests.

—post-data=string

—post-file=file

Use POST as the method for all HTTP requests and send the specified data in the request body. `--post-data` sends *string* as data, whereas `--post-file` sends the contents of *file*. Other than that, they work in exactly the same way.

Please be aware that Wget needs to know the size of the POST data in advance. Therefore the argument to `--post-file` must be a regular file; specifying a FIFO or something like `/dev/stdin` won't work. It's not quite clear how to work around this limitation inherent in HTTP/1.0. Although HTTP/1.1 introduces *chunked* transfer that doesn't require knowing the request length in advance, a client can't use chunked unless it knows it's talking to an HTTP/1.1 server. And it can't know that until it receives a response, which in turn requires the request to have been completed — a chicken-and-egg problem.

Note: if Wget is redirected after the POST request is completed, it will not send the POST data to the redirected URL. This is because URLs that process POST often respond with a redirection to a regular page, which does not desire or accept POST. It is not completely clear that this behavior is optimal; if it doesn't work out, it might be changed in the future.

This example shows how to log to a server using POST and then proceed to download the desired pages, presumably only accessible to authorized users:

```
# Log in to the server. This can be done only once.
wget --save-cookies cookies.txt \
      --post-data 'user=foo&password=bar' \
      http://server.com/auth.php

# Now grab the page or pages we care about.
wget --load-cookies cookies.txt \
      -p http://server.com/interesting/article.php
```

If the server is using session cookies to track user authentication, the above will not work because **—save-cookies** will not save them (and neither will browsers) and the *cookies.txt* file will be empty. In that case use **—keep-session-cookies** along with **—save-cookies** to force saving of session cookies.

HTTPS (SSL/TLS) Options

To support encrypted HTTP (HTTPS) downloads, Wget must be compiled with an external SSL library, currently OpenSSL. If Wget is compiled without SSL support, none of these options are available.

—secure-protocol=protocol

Choose the secure protocol to be used. Legal values are **auto**, **SSLv2**, **SSLv3**, and **TLSv1**. If **auto** is used, the SSL library is given the liberty of choosing the appropriate protocol automatically, which is achieved by sending an SSLv2 greeting and announcing support for SSLv3 and TLSv1. This is the default.

Specifying **SSLv2**, **SSLv3**, or **TLSv1** forces the use of the corresponding protocol. This is useful when talking to old and buggy SSL server implementations that make it hard for OpenSSL to choose the correct protocol version. Fortunately, such servers are quite rare.

—no-check-certificate

Don't check the server certificate against the available certificate authorities. Also don't require the URL host name to match the common name presented by the certificate.

As of Wget 1.10, the default is to verify the server's certificate against the recognized certificate authorities, breaking the SSL handshake and aborting the download if the verification fails. Although this provides more secure downloads, it does break interoperability with some sites that worked with previous Wget versions, particularly those using self-signed, expired, or otherwise invalid certificates. This option forces an "insecure" mode of operation that turns the certificate verification errors into warnings and allows you to proceed.

If you encounter "certificate verification" errors or ones saying that "common name doesn't match requested host name", you can use this option to bypass the verification and proceed with the download. *Only use this option if you are otherwise convinced of the site's authenticity, or if you really don't care about the validity of its certificate.* It is almost always a bad idea not to check the certificates when transmitting confidential or important data.

—certificate=file

Use the client certificate stored in *file*. This is needed for servers that are configured to require certificates from the clients that connect to them. Normally a certificate is not required and this switch is optional.

—certificate-type=type

Specify the type of the client certificate. Legal values are **PEM** (assumed by default) and **DER**, also known as **ASN1**.

—private-key=file

Read the private key from *file*. This allows you to provide the private key in a file separate from the certificate.

—private-key-type=type

Specify the type of the private key. Accepted values are **PEM** (the default) and **DER**.

—ca-certificate=file

Use *file* as the file with the bundle of certificate authorities ("CA") to verify the peers. The certificates must be in PEM format.

Without this option Wget looks for CA certificates at the system-specified locations, chosen at OpenSSL installation time.

—ca-directory=directory

Specifies directory containing CA certificates in PEM format. Each file contains one CA certificate, and the file name is based on a hash value derived from the certificate. This is achieved by processing a certificate directory with the `c_rehash` utility supplied with OpenSSL. Using **—ca-directory** is more efficient than **—ca-certificate** when many certificates are installed because it allows Wget to fetch certificates on demand.

Without this option Wget looks for CA certificates at the system-specified locations, chosen at OpenSSL installation time.

—random-file=file

Use *file* as the source of random data for seeding the pseudo-random number generator on systems without `/dev/random`.

On such systems the SSL library needs an external source of randomness to initialize. Randomness may be provided by EGD (see **—egd-file** below) or read from an external source specified by the user. If this option is not specified, Wget looks for random data in `$RANDFILE` or, if that is unset, in `$HOME/.rnd`. If none of those are available, it is likely that SSL encryption will not be usable.

If you're getting the "Could not seed OpenSSL PRNG; disabling SSL." error, you should provide random data using some of the methods described above.

—egd-file=file

Use *file* as the EGD socket. EGD stands for *Entropy Gathering Daemon*, a user-space program that collects data from various unpredictable system sources and makes it available to other programs that might need it. Encryption software, such as the SSL library, needs sources of non-repeating

randomness to seed the random number generator used to produce cryptographically strong keys.

OpenSSL allows the user to specify his own source of entropy using the `RAND_FILE` environment variable. If this variable is unset, or if the specified file does not produce enough randomness, OpenSSL will read random data from EGD socket specified using this option.

If this option is not specified (and the equivalent startup command is not used), EGD is never contacted. EGD is not needed on modern Unix systems that support `/dev/random`.

FTP Options

— **ftp-user=***user*

— **ftp-password=***password*

Specify the username *user* and password *password* on an FTP server. Without this, or the corresponding startup option, the password defaults to `-wget@`, normally used for anonymous FTP.

Another way to specify username and password is in the URL itself. Either method reveals your password to anyone who bothers to run `ps`. To prevent the passwords from being seen, store them in `.wgetrc` or `.netrc`, and make sure to protect those files from other users with `chmod`. If the passwords are really important, do not leave them lying in those files either—edit the files and delete them after Wget has started the download.

— **no-remove-listing**

Don't remove the temporary `.listing` files generated by FTP retrievals. Normally, these files contain the raw directory listings received from FTP servers. Not removing them can be useful for debugging purposes, or when you want to be able to easily check on the contents of remote server directories (e.g. to verify that a mirror you're running is complete).

Note that even though Wget writes to a known filename for this file, this is not a security hole in the scenario of a user making `.listing` a symbolic link to `/etc/passwd` or something and asking `root` to run Wget in his or her directory. Depending on the options used, either Wget will refuse to write to `.listing`, making the globbing/recursion/time-stamping operation fail, or the symbolic link will be deleted and replaced with the actual `.listing` file, or the listing will be written to a `.listing.number` file.

Even though this situation isn't a problem, though, `root` should never run Wget in a non-trusted user's directory. A user could do something as simple as linking `index.html` to `/etc/passwd` and asking `root` to run Wget with `-N` or `-r` so the file will be overwritten.

— **no-glob**

Turn off FTP globbing. Globbing refers to the use of shell-like special characters (*wildcards*), like `*`, `?`, `[` and `]` to retrieve more than one file from the same directory at once, like:

```
wget ftp://gnjilux.srk.fer.hr/*.msg
```

By default, globbing will be turned on if the URL contains a globbing character. This option may be used to turn globbing on or off permanently.

You may have to quote the URL to protect it from being expanded by your shell. Globbing makes Wget look for a directory listing, which is system-specific. This is why it currently works only with Unix FTP servers (and the ones emulating Unix `ls` output).

— **no-passive-ftp**

Disable the use of the *passive* FTP transfer mode. Passive FTP mandates that the client connect to the server to establish the data connection rather than the other way around.

If the machine is connected to the Internet directly, both passive and active FTP should work equally well. Behind most firewall and NAT configurations passive FTP has a better chance of working. However, in some rare firewall configurations, active FTP actually works when passive FTP doesn't. If you suspect this to be the case, use this option, or set `passive_ftp=off` in your init file.

—retr-symlinks

Usually, when retrieving FTP directories recursively and a symbolic link is encountered, the linked-to file is not downloaded. Instead, a matching symbolic link is created on the local filesystem. The pointed-to file will not be downloaded unless this recursive retrieval would have encountered it separately and downloaded it anyway.

When **—retr-symlinks** is specified, however, symbolic links are traversed and the pointed-to files are retrieved. At this time, this option does not cause Wget to traverse symlinks to directories and recurse through them, but in the future it should be enhanced to do this.

Note that when retrieving a file (not a directory) because it was specified on the command-line, rather than because it was recursed to, this option has no effect. Symbolic links are always traversed in this case.

—no-http-keep-alive

Turn off the “keep-alive” feature for HTTP downloads. Normally, Wget asks the server to keep the connection open so that, when you download more than one document from the same server, they get transferred over the same TCP connection. This saves time and at the same time reduces the load on the server.

This option is useful when, for some reason, persistent (keep-alive) connections don’t work for you, for example due to a server bug or due to the inability of server-side scripts to cope with the connections.

Recursive Retrieval Options**—r****—recursive**

Turn on recursive retrieving.

—l *depth***—level=*depth***

Specify recursion maximum depth level *depth*. The default maximum depth is 5.

—delete-after

This option tells Wget to delete every single file it downloads, *after* having done so. It is useful for pre-fetching popular pages through a proxy, e.g.:

```
wget -r -nd --delete-after http://whatever.com/~popular/page/
```

The **—r** option is to retrieve recursively, and **—nd** to not create directories.

Note that **—delete-after** deletes files on the local machine. It does not issue the **DELE** command to remote FTP sites, for instance. Also note that when **—delete-after** is specified, **—convert-links** is ignored, so **.orig** files are simply not created in the first place.

—k**—convert-links**

After the download is complete, convert the links in the document to make them suitable for local viewing. This affects not only the visible hyperlinks, but any part of the document that links to external content, such as embedded images, links to style sheets, hyperlinks to non-HTML content, etc.

Each link will be changed in one of the two ways:

- The links to files that have been downloaded by Wget will be changed to refer to the file they point to as a relative link.

Example: if the downloaded file */foo/doc.html* links to */bar/img.gif*, also downloaded, then the link in *doc.html* will be modified to point to ***../bar/img.gif***. This kind of transformation works reliably for arbitrary combinations of directories.

- The links to files that have not been downloaded by Wget will be changed to include host name and absolute path of the location they point to.

Example: if the downloaded file */foo/doc.html* links to */bar/img.gif* (or to *../bar/img.gif*), then the link in *doc.html* will be modified to point to *http://hostname/bar/img.gif*.

Because of this, local browsing works reliably: if a linked file was downloaded, the link will refer to its local name; if it was not downloaded, the link will refer to its full Internet address rather than presenting a broken link. The fact that the former links are converted to relative links ensures that you can move the downloaded hierarchy to another directory.

Note that only at the end of the download can Wget know which links have been downloaded. Because of that, the work done by **-k** will be performed at the end of all the downloads.

-K

—backup-converted

When converting a file, back up the original version with a **.orig** suffix. Affects the behavior of **-N**.

-m

—mirror

Turn on options suitable for mirroring. This option turns on recursion and time-stamping, sets infinite recursion depth and keeps FTP directory listings. It is currently equivalent to **-r -N -l inf --no-remove-listing**.

-p

—page-requisites

This option causes Wget to download all the files that are necessary to properly display a given HTML page. This includes such things as inlined images, sounds, and referenced stylesheets.

Ordinarily, when downloading a single HTML page, any requisite documents that may be needed to display it properly are not downloaded. Using **-r** together with **-l** can help, but since Wget does not ordinarily distinguish between external and inlined documents, one is generally left with “leaf documents” that are missing their requisites.

For instance, say document *1.html* contains an **** tag referencing *1.gif* and an **<A>** tag pointing to external document *2.html*. Say that *2.html* is similar but that its image is *2.gif* and it links to *3.html*. Say this continues up to some arbitrarily high number.

If one executes the command:

```
wget -r -l 2 http://<site>/1.html
```

then *1.html*, *1.gif*, *2.html*, *2.gif*, and *3.html* will be downloaded. As you can see, *3.html* is without its requisite *3.gif* because Wget is simply counting the number of hops (up to 2) away from *1.html* in order to determine where to stop the recursion. However, with this command:

```
wget -r -l 2 -p http://<site>/1.html
```

all the above files *and* *3.html*'s requisite *3.gif* will be downloaded. Similarly,

```
wget -r -l 1 -p http://<site>/1.html
```

will cause *1.html*, *1.gif*, *2.html*, and *2.gif* to be downloaded. One might think that:

```
wget -r -l 0 -p http://<site>/1.html
```

would download just *1.html* and *1.gif*, but unfortunately this is not the case, because **-l 0** is equivalent to **-l inf**—that is, infinite recursion. To download a single HTML page (or a handful of them, all specified on the command-line or in a **-i** URL input file) and its (or their) requisites, simply leave off **-r** and **-l**:


```
wget -p http://<site>/1.html
```

Note that Wget will behave as if **-r** had been specified, but only that single page and its requisites will be downloaded. Links from that page to external documents will not be followed. Actually, to download a single page and all its requisites (even if they exist on separate websites), and make sure the lot displays properly locally, this author likes to use a few options in addition to **-p**:

```
wget -E -H -k -K -p http://<site>/<document>
```

To finish off this topic, it's worth knowing that Wget's idea of an external document link is any URL specified in an `<A>` tag, an `<AREA>` tag, or a `<LINK>` tag other than `<LINK REL="stylesheet">`.

—strict-comments

Turn on strict parsing of HTML comments. The default is to terminate comments at the first occurrence of `-->`.

According to specifications, HTML comments are expressed as SGML *declarations*. Declaration is special markup that begins with `<!--` and ends with `>`, such as `<!DOCTYPE ...>`, that may contain comments between a pair of `--` delimiters. HTML comments are “empty declarations”, SGML declarations without any non-comment text. Therefore, `<!--foo-->` is a valid comment, and so is `<!--one---two-->`, but `<!--1---2-->` is not.

On the other hand, most HTML writers don't perceive comments as anything other than text delimited with `<!--` and `-->`, which is not quite the same. For example, something like `<!------->` works as a valid comment as long as the number of dashes is a multiple of four (!). If not, the comment technically lasts until the next `--`, which may be at the other end of the document. Because of this, many popular browsers completely ignore the specification and implement what users have come to expect: comments delimited with `<!--` and `-->`.

Until version 1.9, Wget interpreted comments strictly, which resulted in missing links in many web pages that displayed fine in browsers, but had the misfortune of containing non-compliant comments. Beginning with version 1.9, Wget has joined the ranks of clients that implements “naive” comments, terminating each comment at the first occurrence of `-->`.

If, for whatever reason, you want strict comment parsing, use this option to turn it on.

Recursive Accept/Reject Options

-A *acclist* — **accept** *acclist*

-R *rejlist* — **reject** *rejlist*

Specify comma-separated lists of file name suffixes or patterns to accept or reject (@pxref{Types of Files} for more details).

-D *domain-list*

--domains=*domain-list*

Set domains to be followed. *domain-list* is a comma-separated list of domains. Note that it does *not* turn on **-H**.

--exclude-domains *domain-list*

Specify the domains that are *not* to be followed..

--follow-ftp

Follow FTP links from HTML documents. Without this option, Wget will ignore all the FTP links.

--follow-tags=*list*

Wget has an internal table of HTML tag / attribute pairs that it considers when looking for linked documents during a recursive retrieval. If a user wants only a subset of those tags to be considered, however, he or she should be specify such tags in a comma-separated *list* with this option.

—ignore-tags=*list*

This is the opposite of the **—follow-tags** option. To skip certain HTML tags when recursively looking for documents to download, specify them in a comma-separated *list*.

In the past, this option was the best bet for downloading a single page and its requisites, using a command-line like:

```
wget --ignore-tags=a,area -H -k -K -r http://<site>/<document>
```

However, the author of this option came across a page with tags like `<LINK REL="home" HREF="/ ">` and came to the realization that specifying tags to ignore was not enough. One can't just tell Wget to ignore `<LINK>`, because then stylesheets will not be downloaded. Now the best bet for downloading a single page and its requisites is the dedicated **—page-requisites** option.

-H**—span-hosts**

Enable spanning across hosts when doing recursive retrieving.

-L**—relative**

Follow relative links only. Useful for retrieving a specific home page without any distractions, not even those from the same hosts.

-I *list***—include-directories=*list***

Specify a comma-separated list of directories you wish to follow when downloading (@pxref{Directory-Based Limits} for more details.) Elements of *list* may contain wildcards.

-X *list***—exclude-directories=*list***

Specify a comma-separated list of directories you wish to exclude from download (@pxref{Directory-Based Limits} for more details.) Elements of *list* may contain wildcards.

-np**—no-parent**

Do not ever ascend to the parent directory when retrieving recursively. This is a useful option, since it guarantees that only the files *below* a certain hierarchy will be downloaded.

EXAMPLES

The examples are divided into three sections loosely based on their complexity.

Simple Usage

- Say you want to download a URL. Just type:

```
wget http://fly.srk.fer.hr/
```

- But what will happen if the connection is slow, and the file is lengthy? The connection will probably fail before the whole file is retrieved, more than once. In this case, Wget will try getting the file until it either gets the whole of it, or exceeds the default number of retries (this being 20). It is easy to change the number of tries to 45, to insure that the whole file will arrive safely:

```
wget --tries=45 http://fly.srk.fer.hr/jpg/flyweb.jpg
```

- Now let's leave Wget to work in the background, and write its progress to log file *log*. It is tiring to type **—tries**, so we shall use **-t**.

```
wget -t 45 -o log http://fly.srk.fer.hr/jpg/flyweb.jpg &
```

The ampersand at the end of the line makes sure that Wget works in the background. To unlimited the number of retries, use **-t inf**.

- The usage of FTP is as simple. Wget will take care of login and password.

```
wget ftp://gnjilux.srk.fer.hr/welcome.msg
```

- If you specify a directory, Wget will retrieve the directory listing, parse it and convert it to HTML. Try:

```
wget ftp://ftp.gnu.org/pub/gnu/
links index.html
```

Advanced Usage

- You have a file that contains the URLs you want to download? Use the **-i** switch:

```
wget -i <file>
```

If you specify - as file name, the URLs will be read from standard input.

- Create a five levels deep mirror image of the GNU web site, with the same directory structure the original has, with only one try per document, saving the log of the activities to *gnulog*:

```
wget -r http://www.gnu.org/ -o gnulog
```

- The same as the above, but convert the links in the HTML files to point to local files, so you can view the documents off-line:

```
wget --convert-links -r http://www.gnu.org/ -o gnulog
```

- Retrieve only one HTML page, but make sure that all the elements needed for the page to be displayed, such as inline images and external style sheets, are also downloaded. Also make sure the downloaded page references the downloaded links.

```
wget -p --convert-links http://www.server.com/dir/page.html
```

The HTML page will be saved to *www.server.com/dir/page.html*, and the images, stylesheets, etc., somewhere under *www.server.com/*, depending on where they were on the remote server.

- The same as the above, but without the *www.server.com/* directory. In fact, I don't want to have all those random server directories anyway---just save *all* those files under a *download/* subdirectory of the current directory.

```
wget -p --convert-links -nH -nd -Pdownload \
http://www.server.com/dir/page.html
```

- Retrieve the index.html of **www.lycos.com**, showing the original server headers:

```
wget -S http://www.lycos.com/
```

- Save the server headers with the file, perhaps for post-processing.

```
wget --save-headers http://www.lycos.com/
more index.html
```

- Retrieve the first two levels of **wuarchive.wustl.edu**, saving them to */tmp*.

```
wget -r -l2 -P/tmp ftp://wuarchive.wustl.edu/
```

- You want to download all the GIFs from a directory on an HTTP server. You tried **wget http://www.server.com/dir/*.gif**, but that didn't work because HTTP retrieval does not support globbing. In that case, use:

```
wget -r -l1 --no-parent -A.gif http://www.server.com/dir/
```

More verbose, but the effect is the same. **-r -l1** means to retrieve recursively, with maximum depth of 1. **--no-parent** means that references to the parent directory are ignored, and **-A.gif** means to download only the GIF files. **-A "*.gif"** would have worked too.

- Suppose you were in the middle of downloading, when Wget was interrupted. Now you do not want to clobber the files already present. It would be:

```
wget -nc -r http://www.gnu.org/
```

- If you want to encode your own username and password to HTTP or FTP, use the appropriate URL syntax.

```
wget ftp://hnksic:mypassword@unix.server.com/.emacs
```

Note, however, that this usage is not advisable on multi-user systems because it reveals your password to anyone who looks at the output of `ps`.

- You would like the output documents to go to standard output instead of to files?

```
wget -O - http://jagor.srce.hr/ http://www.srce.hr/
```

You can also combine the two options and make pipelines to retrieve the documents from remote hotlists:

```
wget -O - http://cool.list.com/ | wget --force-html -i -
```

Very Advanced Usage

- If you wish Wget to keep a mirror of a page (or FTP subdirectories), use **--mirror (-m)**, which is the shorthand for **-r -l inf -N**. You can put Wget in the crontab file asking it to recheck a site each Sunday:

```
crontab
0 0 * * 0 wget --mirror http://www.gnu.org/ -o /home/me/weeklog
```

- In addition to the above, you want the links to be converted for local viewing. But, after having read this manual, you know that link conversion doesn't play well with timestamping, so you also want Wget to back up the original HTML files before the conversion. Wget invocation would look like this:

```
wget --mirror --convert-links --backup-converted \
http://www.gnu.org/ -o /home/me/weeklog
```

- But you've also noticed that local viewing doesn't work all that well when HTML files are saved under extensions other than **.html**, perhaps because they were served as *index.cgi*. So you'd like Wget to rename all the files served with content-type **text/html** or **application/xhtml+xml** to *name.html*.

```
wget --mirror --convert-links --backup-converted \
    --html-extension -o /home/me/weeklog \
    http://www.gnu.org/
```

Or, with less typing:

```
wget -m -k -K -E http://www.gnu.org/ -o /home/me/weeklog
```

FILES

/usr/local/etc/wgetrc

Default location of the *global* startup file.

.wgetrc

User startup file.

BUGS

You are welcome to send bug reports about GNU Wget to **<bug-wget@gnu.org>**.

Before actually submitting a bug report, please try to follow a few simple guidelines.

1. Please try to ascertain that the behavior you see really is a bug. If Wget crashes, it's a bug. If Wget does not behave as documented, it's a bug. If things work strange, but you are not sure about the way they are supposed to work, it might well be a bug.
2. Try to repeat the bug in as simple circumstances as possible. E.g. if Wget crashes while downloading **wget -r10 -kKE -t5 -Y0 http://yoyodyne.com -o /tmp/log**, you should try to see if the crash is repeatable, and if will occur with a simpler set of options. You might even try to start the download at the page where the crash occurred to see if that page somehow triggered the crash.

Also, while I will probably be interested to know the contents of your *.wgetrc* file, just dumping it into the debug message is probably a bad idea. Instead, you should first try to see if the bug repeats with *.wgetrc* moved out of the way. Only if it turns out that *.wgetrc* settings affect the bug, mail me the relevant parts of the file.

3. Please start Wget with **-d** option and send us the resulting output (or relevant parts thereof). If Wget was compiled without debug support, recompile it—it is *much* easier to trace bugs with debug support on.

Note: please make sure to remove any potentially sensitive information from the debug log before sending it to the bug address. The **-d** won't go out of its way to collect sensitive information, but the log *will* contain a fairly complete transcript of Wget's communication with the server, which may include passwords and pieces of downloaded data. Since the bug address is publically archived, you may assume that all bug reports are visible to the public.

4. If Wget has crashed, try to run it in a debugger, e.g. `gdb `which wget` core` and type `where` to get the backtrace. This may not work if the system administrator has disabled core files, but it is safe to try.

SEE ALSO

GNU Info entry for *wget*.

AUTHOR

Originally written by Hrvoje Niksic <hnksic@xemacs.org>.

COPYRIGHT

Copyright (c) 1996 — 2005 Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being “GNU General Public License” and “GNU Free Documentation License”,

with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

NAME

whatis – search the whatis database for complete words

SYNOPSIS

edrc/bin/whatis *keyword* ...

AVAILABILITY

WA2L/edrc

DESCRIPTION

whatis searches a set of database files containing short descriptions of system commands for keywords and displays the result on the standard output. Only complete word matches are displayed.

OPTIONS

keyword keyword to be searched.

FILES

whatis whatis databases in the \$MANPATH.

SEE ALSO

edrcintro(1), **apropos(1)**, **edrcman(1)**, **man(1)**

NOTES

Parts of this manpage were extracted from the documentation of the whatis man page written by John W. Eaton and modified to fit into WA2L/edrc package.

BUGS

-

AUTHOR

whatis was developed by John W. Eaton, <jwe@che.utexas.edu>, Department of Chemical Engineering, The University of Texas at Austin, Austin Texas 78712 and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

whereami – print server environment/system information

SYNOPSIS

edrc/bin/whereami [**-h**]

whereami [**-l**] [**-u** *username*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print the server environment name and the name of the related customer where logged on.

In addition the structured user definition information of the current user can be printed.

This command is intended for interactive human information. Basically to display where logged on. To compute this information in other scripts, use the **server_environment**(3) and/or the **user_info**(3) command.

OPTIONS

-h usage message.

-l long output. The long output extends the normal output with the user account information details.

-u *username*
name of the user account whose detail information should be printed if the information of a user account that differs from the user that executes the **whereami** command.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 operating system is not supported, yet. See **osid**(3) if you get this error.

4 usage printed.

FILES

-

EXAMPLES

1) common usage

The following output is returned when logged on to a system that is used as part of the *PRODUCTION* environment for the customer *ACME*.

```
[ / ]  
[ jdoe@acme001 ][bash]: whereami
```

LOGGED ON TO SERVER ENVIRONMENT:

| | |
|-------------|--------------------------------------|
| Customer | : ACME |
| Environment | : PRODUCTION |
| Description | : ACME Corp. Production, Switzerland |

SEE ALSO

edrcintro(1), **server_environment(3)**, **user_info(3)**

NOTES

-

BUGS

-

AUTHOR

whereami was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO

WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

whoisin – list users that are logged in to remote systems

SYNOPSIS

edrc/bin/whoisin [**-h**] [*hostlist*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

print a list of users who are logged in on certain (remote) hosts.

The connect to the remote systems is performed using **ssh-exec** internally; the resolution of logged on users is done using **who**(1).

OPTIONS

-h usage message.

hostlist list of hosts to list the users.

The *hostlist* may contain *@HOSTGRPS* and normal *hostnames*.

If no *hostlist* is defined, the hosts to check are read from the **etc/whoisin.list** file.

ENVIRONMENT

-

EXIT STATUS

4 usage printed.

0 no error.

FILES

edrc/etc/whoisin.list

list of hosts to be checked when no *hostlist* is specified on the command line.

EXAMPLES

-

SEE ALSO

edrcintro(1), **hostlist(3)**, **resolve_targetlist(3)**, **ssh-exec(1)**, **who(1)**, **whoisin.list(4)**

NOTES

-

BUGS

-

AUTHOR

whoisin was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

whoisin.list – host list definition for whoisin

SYNOPSIS

edrc/etc/whoisin.list

AVAILABILITY

WA2L/edrc

DESCRIPTION

This is the target host definition for the **whoisin** command.

FILEFORMAT

The fileformat is a plain list of hostnames and hostgroups.

Rows starting with a **#** are considered as comments. Empty lines are allowed, too.

OPTIONS

-

EXAMPLES

```
#
# whoisin.list - configuration file for whoisin
#
# [00] 06.01.2000 CWa Initial Version
#

acme-004 acme-005 acme-006 @ACME.APPS
lab-001 lab-003 @TEST
```

SEE ALSO

edrcintro(1), **hostlist(4)**, **resolve_targetlist(3)**, **whoisin(1)**

NOTES

The entries specified in this configuration file are resolved by **resolve_targetlist(3)**.

BUGS

-

AUTHOR

whoisin.list was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

winexe – Winexe is a Remote Windows–command executor

SYNOPSIS

```
winexe [-h] [-V] [-U [DOMAIN/]USERNAME[%PASSWORD]] [-A FILE] [-N] [-k]
      [-d DEBUGLEVEL] [--uninstall] [--reinstall] [--runas [DOMAIN/]USERNAME[%PASSWORD]]
      [--runas-file FILE] [--interactive [0|1]] [--ostype [0|1]]
```

DESCRIPTION

This tool is part of the **samba**(7) suite.

The winexe allows remote command execution on native Windows operating systems.

OPTIONS

-?|--help
Print a summary of command line options.

-V|--version
Prints the program version number.

-U|--user=username[%password]
Sets the SMB username or username and password.

If %password is not specified, the user will be prompted. The client will first check the **USER** environment variable, then the **LOGNAME** variable and if either exists, the string is uppercased. If these environmental variables are not found, the username **GUEST** is used.

A third option is to use a credentials file which contains the plaintext of the username and password. This option is mainly provided for scripts where the admin does not wish to pass the credentials on the command line or via environment variables. If this method is used, make certain that the permissions on the file restrict access from unwanted users. See the **-A** for more details.

Be cautious about including passwords in scripts. Also, on many systems the command line of a running process may be seen via the ps command. To be safe always allow rpcclient to prompt for a password and type it in directly.

-A|--authentication-file=filename
This option allows you to specify a file from which to read the username and password used in the connection. The format of the file is

```
username = <value>
password = <value>
domain   = <value>
```

Make certain that the permissions on the file restrict access from unwanted users.

-N|--no-pass
If specified, this parameter suppresses the normal password prompt from the client to the user. This is useful when accessing a service that does not require a password.

Unless a password is specified on the command line or this parameter is specified, the client will request a password.

If a password is specified on the command line and this option is also defined the password on the command line will be silently ignored and no password will be used.

-k|--kerberos
Try to authenticate with kerberos. Only useful in an Active Directory environment.

-d|--debuglevel=level

level is an integer from 0 to 10. The default value if this parameter is not specified is 1.

The higher this value, the more detail will be logged to the log files about the activities of the server. At level 0, only critical errors and serious warnings will be logged. Level 1 is a reasonable level for day-to-day running – it generates a small amount of information about operations carried out.

Levels above 1 will generate considerable amounts of log data, and should only be used when investigating a problem. Levels above 3 are designed for use only by developers and generate HUGE amounts of log data, most of which is extremely cryptic.

Note that specifying this parameter here will override the **log level** parameter in the smb.conf file.

--uninstall

Uninstall winexe service after remote execution.

--reinstall

Reinstall winexe service before remote execution.

--runas [DOMAIN/]USERNAME[%PASSWORD]]

Run as the given user (BEWARE: this password is sent in cleartext over the network!)

--runas-file FILE

Run as user options defined in a file.

--interactive [0|1]

Desktop interaction.

There are two options:

- 0 – disallow
- 1 – allow. If allow, also use the **---system** switch (Windows requirement). Vista does not support this option.

--ostype [0|1|2]

Determines which version (32-bit or 64-bit) of service will be installed.

There are three options:

- 0 – 32-bit
- 1 – 64-bit
- 2 – winexe will decide

EXIT STATUS

The winexe program returns 0 if the operation succeeded, or 1 if the operation failed.

VERSION

This man page is part of version 4.13.2 of the Samba suite.

AUTHOR

The original Samba software and related utilities were created by Andrew Tridgell. Samba is now developed by the Samba Team as an Open Source project similar to the way the Linux kernel is developed.

The winexe and its native Windows counterpart were written by Andrzej Hajda. The Samba client tool winexe was later rewritten by Volker Lendecke.

This manpage was written by Guenther Deschner.

NAME

wmic – WMI client

SYNOPSIS

```
[ -? | --help ] [ --usage ] [ -d | --debuglevel DEBUGLEVEL ] [ --debug-stderr ] [ -s | --configfile CONFIGFILE ]
[ --option=name=value ] [ -l | --log-basename LOGFILEBASE ] [ --leak-report ]
[ --leak-report-full ] [ -R | --name-resolve NAME-RESOLVE-ORDER ] [ -O | --socket-options SOCKETOPTIONS ]
[ -n | --netbiosname name NETBIOSNAME ] [ -W | --workgroup WORKGROUP ]
[ --realm=REALM ] [ -i | --scope SCOPE ] [ -m | --maxprotocol MAXPROTOCOL ] [ -U | --user [DOMAIN]USERNAME[%PASSWORD] ]
[ -N | --no-pass ] [ --password=STRING ] [ -A | --authentication-file FILE ] [ -S | --signing on|off|required ] [ -P | --machine-pass ] [ --simple-bind-dn=STRING ]
[ -k | --kerberos STRING ] [ --use-security-mechanisms=STRING ] [ -V | --version ] [ --namespace=STRING ] //host query
```

EXAMPLE

Example: wmic -U [domain/]adminuser%password //host "select * from Win32_ComputerSystem"

DESCRIPTION

--namespace=STRING

WMI namespace, default to root\cimv2

Help options:

-?, --help

Show this help message

--usage

Display brief usage message

Common samba options:

-d, --debuglevel=DEBUGLEVEL

Set debug level

--debug-stderr

Send debug output to STDERR

-s, --configfile=CONFIGFILE

Use alternative configuration file

--option=name=value

Set smb.conf option from command line

-l, --log-basename=LOGFILEBASE

Basename for log/debug files

--leak-report

enable talloc leak reporting on exit

--leak-report-full

enable full talloc leak reporting on exit

Connection options:

-R, --name-resolve=NAME-RESOLVE-ORDER

Use these name resolution services only

-O, --socket-options=SOCKETOPTIONS

socket options to use

-n, --netbiosname=NETBIOSNAME

Primary netbios name

-W, --workgroup=WORKGROUP

Set the workgroup name

- realm=REALM**
Set the realm name
- i, --scope=SCOPE**
Use this Netbios scope
- m, --maxprotocol=MAXPROTOCOL**
Set max protocol level

Authentication options:

- U, --user=[DOMAIN]USERNAME[%PASSWORD]**
Set the network username
- N, --no-pass**
Don't ask for a password
- password=STRING**
Password
- A, --authentication-file=FILE**
Get the credentials from a file
- S, --signing=on|off|required**
Set the client signing state
- P, --machine-pass**
Use stored machine account password (implies **-k**)
- simple-bind-dn=STRING**
DN to use for a simple bind
- k, --kerberos=STRING**
Use Kerberos
- use-security-mechanisms=STRING**
Restricted list of authentication mechanisms available for use with this authentication

Common samba options:

- V, --version**
Print version

EXAMPLE

```
wmic -U [domain/]adminuser%password //host "select * from Win32_ComputerSystem"
```

AUTHOR

This manual page was written by Bernd Zeimetz <bzed@debian.org> for the Debian project (but may be used by others).

NAME

woist – Search a Path For a File

SYNOPSIS

edrc/bin/woist [**-a**] [**-xdfhrs**] [**-p** *directory-list*] *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

does what whereis should have done. It examines a list of directories and looks for a file whose attributes match the corresponding arguments. The default is to look on \$PATH for executable files, but any of the standard arguments to **test** may be given, so that it is possible to look for a readable file, a writable file, or a non-empty file on any arbitrary path. If the file is found in one of directories in the path, the files path is printed to stdout.

If the file name contains a slash ("/"), the directory-list is ignored. The specified file is checked according to the conditions; if it exists and matches, its path is printed to stdout.

OPTIONS

-a All - list all instances found in all directories on the path. Otherwise only print the first one found.

-dfhrsx **test** argument. Only one may be specified. Options are:

-d Directory

-f Existing File of Any Type

-h Symbolic Link

-r Readable file

-s Non-Empty File

-w Writable File

-x Executable File (default)

-p '*directory-list*'

List of directories to examine. Default is \$PATH, but any colon separated list of directories may be used. The list is echoed by csh(1), to ensure that tilde-username in a directory have been properly resolved.

file File to be found.

EXIT STATUS

| | |
|----------|--------------------------------|
| 0 | file found |
| 1 | error |
| 4 | usage listed or file not found |

EXAMPLES

-

SEE ALSO

edrcintro(1), **whereis(1)**

FILES

-

NOTES

Parts of this manpage were extracted from the documentation of the **SFI-Director** and modified to fit to the **WA2L/edrc** package. See <http://sfidirector.sourceforge.net> for more information about the **SFI-Director**.

BUGS

-

AUTHOR

woist was developed by Peter Stevens, SFI and integrated into **WA2L/edrc** by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

xbdf – multiple platform bdf / df -k

SYNOPSIS

edrc/bin/xbdf [**-h**]

xbdf [**-l**] [*filesystem* | *file*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

a platform independent version of the **bdf** respectively **df -k** command. Line breaks are removed, as experienced on HP-UX in certain circumstances.

OPTIONS

-h usage message.

-l display only local file systems.

filesystem file system to display the sizes from.

file show information about the file system on which the *file* resides.

ENVIRONMENT

-

EXIT STATUS

0 no error.

2 operating system not supported. See **osid(3)** if you get this error.

4 usage displayed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **df(1)**, **bdf(1)**

NOTES

-

BUGS

-

AUTHOR

xbdf was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2009 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

xlog – write stdin to log with a timestamp

SYNOPSIS

edrc/bin/xlog [**-h**]

xlog [**-t tag**] **-f logfile**

AVAILABILITY

WA2L/edrc

DESCRIPTION

write data from **stdin** to a *logfile*.

Each line is preceded with a date, time, sequence and tag entry.

A common use might be in crontab instead of piping outputs to **/dev/null**.

Example:

```
a_command 2>&1 | xlog -t a_cmd -f /var/adm/log/a_command.log
```

OPTIONS

-h usage message.

-t tag tag to be added to the log file.

-f logfile log file name.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 cannot write to *logfile*.

2 operating system not supported.

4 usage printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **log(3)**, **tscat(3)**, **xtee(1)**

NOTES

-

BUGS

-

AUTHOR

xlog was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME**SYNOPSIS**

edrc/lib/xml2csv [**-h**]

xml2csv [{ **-l** *fieldname* | **-k** *fieldlist* | **-i** *fieldlist* | **-e** *emptydata* | **-s** *separator* }] (*sourcefile* | *URL*) *destinationfile*

AVAILABILITY

WA2L/edrc

DESCRIPTION

Command line XML to CSV converter.

xml2csv is command line tool for converting data from XML schema to CSV. The tool has many command line options. The software is platform independent and was written in Java language.

Features:

- Converts XML schema to CSV file
- Can deal with filenames and urls
- Automatically detects loops (repeated elements) in XML used for splitting data to rows
- Allows to override name of the loop/repeated field
- Allows to keep only specific tags/fields
- Allows to ignore specific tags/fields
- Allows to set values for empty data and CSV separator
- Supports distinct option
- Platform Independent

OPTIONS

-h usage message.

-l *fieldname* set the name of the field that repeats in XML schema.

-k *fieldlist* comma separated list of fields to be kept.

-i *fieldlist* comma separated list of fields to be ignored.

-e *emptydata* data to be used for empty fields.

-s *separator*
field separator in **CSV** file.

sourcefile | *URL*
file or url to read the **XML** data from.

destinationfile
CSV file to write the data to.

ENVIRONMENT

-

EXIT STATUS

0 always.

FILES

-

EXAMPLES

Example usage of the **xml2csv** command:

```
xml2csv -l field -i city,country -d -s "," data.xml data.csv
xml2csv -k "name, surname" data.xml data.csv
xml2csv http://www.example.com/data.xml data.csv
xml2csv data.xml data.csv
```

SEE ALSO

edrcintro(1), **csvcat(3)**, **print_list(3)**, **select_columns(3)**, **xml2json(3)**, <https://code.google.com/archive/p/xml2csv-conv>

NOTES

parts of the documentation are extracted from the commands usage message and the authors web page (<https://code.google.com/archive/p/xml2csv-conv>) and were integrated into WA2L/edrc.

BUGS

-

AUTHOR

xml2csv-conv was developed by an unknown author (<https://code.google.com/archive/p/xml2csv-conv>) and integrated as xml2csv into WA2L/edrc by Christian Walther. Send suggestions and bug reports regarding to the integration to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME**SYNOPSIS**

edrc/lib/xml2json *file.xml*

xml2json < *file.xml*

AVAILABILITY

WA2L/edrc

DESCRIPTION

Command line **XML** to **JSON** converter.

xml2json is command line tool for converting data from **XML** schema to **JSON**.

OPTIONS

file.xml the **XML** file to be converted to **JSON**.

ENVIRONMENT

-

EXIT STATUS

0 no error.

1 error.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **jq**(3), **xml2csv**(3), <http://github.com/Cheedoong/xml2json/>

NOTES

-

BUGS

-

AUTHOR

xml2json was developed by Cheedoong (<http://github.com/Cheedoong/xml2json/>) and integrated into WA2L/edrc by Christian Walther. Send suggestions and bug reports regarding to the integration to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2021 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

xmore – show files page-wise in recovery scripts

SYNOPSIS

edrc/bin/xmore [**-h**]

xmore *file...*

AVAILABILITY

WA2L/edrc

DESCRIPTION

display file(s) page by page in WA2L/edrc recovery scripts.

The **more**(1) or **less**(1) command do not stop for user input after displaying a page if used in a recovery scripts, but the **xmore** command does.

OPTIONS

-h usage message.

file... list of file(s) to be displayed page-wise in recovery scripts.

ENVIRONMENT

-

EXIT STATUS

4 usage printed.

0 no error.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrcscripts(1m)**, **more(1)**, **less(1)**

NOTES

-

BUGS

-

AUTHOR

xmore was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

xpid – select proccmd from process list (enhanced ps -ef | grep pattern)

SYNOPSIS

edrc/bin/xpid [*proccmd-pattern*]

AVAILABILITY

WA2L/edrc

DESCRIPTION

select processes from the process command part of the process list.

This is a convenience command to the **pslist**(1) command.

Internally the **xpid** command calls currently **pslist -l -c '.*selection.*'** .

The **xpid** command is intended for manual command line use only, for script use the **pslist** command should be used because the output of the **xpid** command is not intended for further processing and might be changed.

OPTIONS

proccmd-pattern

regular expression to select only from the proccmd column of the process list.

ENVIRONMENT

-

EXIT STATUS

x exit status of **pslist**.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **ps(1)**, **pslist(3)**

NOTES

-

BUGS

-

AUTHOR

xpid was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

xtee – tee with timestamp output to file

SYNOPSIS

edrc/bin/xtee

xtee *file*

AVAILABILITY

WA2L/edrc

DESCRIPTION

xtee(1) an enhanced version of the standard **tee** command.

This one appends the standard output with extra data and session columns to the logfile.

OPTIONS

usage message.

file file to append the data received thru **stdin**.

ENVIRONMENT

-

EXIT STATUS

4 usage printed.

0 no error.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **edrc(1m)**, **tscat(1)**, **tee(1)**

NOTES

-

BUGS

The **xtee(1)** command causes a SIGSEGV signal on HP-UX 11.00 in some circumstances. Therefore it is currently not used in the **edrc** command. To save the output to the logfiles **tee(1)** is used as long as this issue is not resolved completely. On Linux and Solaris this behavior was not discovered so far.

AUTHOR

xtee was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net

.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

xz, **unxz**, **xzcat**, **lzma**, **unlzma**, **lzcat** – Compress or decompress **.xz** and **.lzma** files

SYNOPSIS

xz [*option...*] [*file...*]

COMMAND ALIASES

unxz is equivalent to **xz --decompress**.

xzcat is equivalent to **xz --decompress --stdout**.

lzma is equivalent to **xz --format=lzma**.

unlzma is equivalent to **xz --format=lzma --decompress**.

lzcat is equivalent to **xz --format=lzma --decompress --stdout**.

When writing scripts that need to decompress files, it is recommended to always use the name **xz** with appropriate arguments (**xz -d** or **xz -dc**) instead of the names **unxz** and **xzcat**.

DESCRIPTION

xz is a general-purpose data compression tool with command line syntax similar to **gzip(1)** and **bzip2(1)**. The native file format is the **.xz** format, but the legacy **.lzma** format used by LZMA Utils and raw compressed streams with no container format headers are also supported.

xz compresses or decompresses each *file* according to the selected operation mode. If no *files* are given or *file* is **-**, **xz** reads from standard input and writes the processed data to standard output. **xz** will refuse (display an error and skip the *file*) to write compressed data to standard output if it is a terminal. Similarly, **xz** will refuse to read compressed data from standard input if it is a terminal.

Unless **--stdout** is specified, *files* other than **-** are written to a new file whose name is derived from the source *file* name:

- When compressing, the suffix of the target file format (**.xz** or **.lzma**) is appended to the source filename to get the target filename.
- When decompressing, the **.xz** or **.lzma** suffix is removed from the filename to get the target filename. **xz** also recognizes the suffixes **.txz** and **.tlz**, and replaces them with the **.tar** suffix.

If the target file already exists, an error is displayed and the *file* is skipped.

Unless writing to standard output, **xz** will display a warning and skip the *file* if any of the following applies:

- *File* is not a regular file. Symbolic links are not followed, and thus they are not considered to be regular files.
- *File* has more than one hard link.
- *File* has **setuid**, **setgid**, or sticky bit set.
- The operation mode is set to compress and the *file* already has a suffix of the target file format (**.xz** or **.txz** when compressing to the **.xz** format, and **.lzma** or **.tlz** when compressing to the **.lzma** format).
- The operation mode is set to decompress and the *file* doesn't have a suffix of any of the supported file formats (**.xz**, **.txz**, **.lzma**, or **.tlz**).

After successfully compressing or decompressing the *file*, **xz** copies the owner, group, permissions, access time, and modification time from the source *file* to the target file. If copying the group fails, the permissions are modified so that the target file doesn't become accessible to users who didn't have permission to access the source *file*. **xz** doesn't support copying other metadata like access control lists or extended attributes yet.

Once the target file has been successfully closed, the source *file* is removed unless **--keep** was specified. The source *file* is never removed if the output is written to standard output.

Sending **SIGINFO** or **SIGUSR1** to the **xz** process makes it print progress information to standard error. This has only limited use since when standard error is a terminal, using **--verbose** will display an automatically updating progress indicator.

Memory usage

The memory usage of **xz** varies from a few hundred kilobytes to several gigabytes depending on the compression settings. The settings used when compressing a file determine the memory requirements of the decompressor. Typically the decompressor needs 5 % to 20 % of the amount of memory that the compressor needed when creating the file. For example, decompressing a file created with **xz -9** currently requires 65 MiB of memory. Still, it is possible to have **.xz** files that require several gigabytes of memory to decompress.

Especially users of older systems may find the possibility of very large memory usage annoying. To prevent uncomfortable surprises, **xz** has a built-in memory usage limiter, which is disabled by default. While some operating systems provide ways to limit the memory usage of processes, relying on it wasn't deemed to be flexible enough (e.g. using **ulimit(1)** to limit virtual memory tends to cripple **mmap(2)**).

The memory usage limiter can be enabled with the command line option **--memlimit=limit**. Often it is more convenient to enable the limiter by default by setting the environment variable **XZ_DEFAULTS**, e.g. **XZ_DEFAULTS=--memlimit=150MiB**. It is possible to set the limits separately for compression and decompression by using **--memlimit-compress=limit** and **--memlimit-decompress=limit**. Using these two options outside **XZ_DEFAULTS** is rarely useful because a single run of **xz** cannot do both compression and decompression and **--memlimit=limit** (or **-M limit**) is shorter to type on the command line.

If the specified memory usage limit is exceeded when decompressing, **xz** will display an error and decompressing the file will fail. If the limit is exceeded when compressing, **xz** will try to scale the settings down so that the limit is no longer exceeded (except when using **--format=raw** or **--no-adjust**). This way the operation won't fail unless the limit is very small. The scaling of the settings is done in steps that don't match the compression level presets, e.g. if the limit is only slightly less than the amount required for **xz -9**, the settings will be scaled down only a little, not all the way down to **xz -8**.

Concatenation and padding with .xz files

It is possible to concatenate **.xz** files as is. **xz** will decompress such files as if they were a single **.xz** file.

It is possible to insert padding between the concatenated parts or after the last part. The padding must consist of null bytes and the size of the padding must be a multiple of four bytes. This can be useful e.g. if the **.xz** file is stored on a medium that measures file sizes in 512-byte blocks.

Concatenation and padding are not allowed with **.lzma** files or raw streams.

OPTIONS

Integer suffixes and special values

In most places where an integer argument is expected, an optional suffix is supported to easily indicate large integers. There must be no space between the integer and the suffix.

KiB Multiply the integer by 1,024 (2^{10}). **Ki**, **k**, **kB**, **K**, and **KB** are accepted as synonyms for **KiB**.

MiB Multiply the integer by 1,048,576 (2^{20}). **Mi**, **m**, **M**, and **MB** are accepted as synonyms for **MiB**.

GiB Multiply the integer by 1,073,741,824 (2^{30}). **Gi**, **g**, **G**, and **GB** are accepted as synonyms for **GiB**.

The special value **max** can be used to indicate the maximum integer value supported by the option.

Operation mode

If multiple operation mode options are given, the last one takes effect.

-z, --compress

Compress. This is the default operation mode when no operation mode option is specified and no other operation mode is implied from the command name (for example, **unxz** implies **--decompress**).

-d, --decompress, --uncompress

Decompress.

-t, --test

Test the integrity of compressed *files*. This option is equivalent to **--decompress --stdout** except that the decompressed data is discarded instead of being written to standard output. No files are created or removed.

-l, --list

Print information about compressed *files*. No uncompressed output is produced, and no files are created or removed. In list mode, the program cannot read the compressed data from standard input or from other unseekable sources.

The default listing shows basic information about *files*, one file per line. To get more detailed information, use also the **--verbose** option. For even more information, use **--verbose** twice, but note that this may be slow, because getting all the extra information requires many seeks. The width of verbose output exceeds 80 characters, so piping the output to e.g. **less -S** may be convenient if the terminal isn't wide enough.

The exact output may vary between **xz** versions and different locales. For machine-readable output, **--robot --list** should be used.

Operation modifiers**-k, --keep**

Don't delete the input files.

-f, --force

This option has several effects:

- If the target file already exists, delete it before compressing or decompressing.
- Compress or decompress even if the input is a symbolic link to a regular file, has more than one hard link, or has the setuid, setgid, or sticky bit set. The setuid, setgid, and sticky bits are not copied to the target file.
- When used with **--decompress --stdout** and **xz** cannot recognize the type of the source file, copy the source file as is to standard output. This allows **xzcat --force** to be used like **cat(1)** for files that have not been compressed with **xz**. Note that in future, **xz** might support new compressed file formats, which may make **xz** decompress more types of files instead of copying them as is to standard output. **--format=format** can be used to restrict **xz** to decompress only a single file format.

-c, --stdout, --to-stdout

Write the compressed or decompressed data to standard output instead of a file. This implies **--keep**.

--single-stream

Decompress only the first **.xz** stream, and silently ignore possible remaining input data following the stream. Normally such trailing garbage makes **xz** display an error.

xz never decompresses more than one stream from **.lzma** files or raw streams, but this option still makes **xz** ignore the possible trailing data after the **.lzma** file or raw stream.

This option has no effect if the operation mode is not **--decompress** or **--test**.

--no-sparse

Disable creation of sparse files. By default, if decompressing into a regular file, **xz** tries to make the file sparse if the decompressed data contains long sequences of binary zeros. It also works when writing to standard output as long as standard output is connected to a regular file and certain additional conditions are met to make it safe. Creating sparse files may save disk space and speed up the decompression by reducing the amount of disk I/O.

-S .suf, --suffix=.suf

When compressing, use *.suf* as the suffix for the target file instead of **.xz** or **.lzma**. If not writing to standard output and the source file already has the suffix *.suf*, a warning is displayed and the file is skipped.

When decompressing, recognize files with the suffix *.suf* in addition to files with the **.xz**, **.txz**, **.lzma**, or **.tlz** suffix. If the source file has the suffix *.suf*, the suffix is removed to get the target filename.

When compressing or decompressing raw streams (**--format=raw**), the suffix must always be specified unless writing to standard output, because there is no default suffix for raw streams.

--files[=*file*]

Read the filenames to process from *file*; if *file* is omitted, filenames are read from standard input. Filenames must be terminated with the newline character. A dash (-) is taken as a regular filename; it doesn't mean standard input. If filenames are given also as command line arguments, they are processed before the filenames read from *file*.

--files0[=*file*]

This is identical to **--files[=*file*]** except that each filename must be terminated with the null character.

Basic file format and compression options

-F *format*, --format=*format*

Specify the file *format* to compress or decompress:

auto This is the default. When compressing, **auto** is equivalent to **xz**. When decompressing, the format of the input file is automatically detected. Note that raw streams (created with **--format=raw**) cannot be auto-detected.

xz Compress to the **.xz** file format, or accept only **.xz** files when decompressing.

lzma, alone

Compress to the legacy **.lzma** file format, or accept only **.lzma** files when decompressing. The alternative name **alone** is provided for backwards compatibility with LZMA Utils.

raw Compress or uncompress a raw stream (no headers). This is meant for advanced users only. To decode raw streams, you need use **--format=raw** and explicitly specify the filter chain, which normally would have been stored in the container headers.

-C *check*, --check=*check*

Specify the type of the integrity check. The check is calculated from the uncompressed data and stored in the **.xz** file. This option has an effect only when compressing into the **.xz** format; the **.lzma** format doesn't support integrity checks. The integrity check (if any) is verified when the **.xz** file is decompressed.

Supported *check* types:

none Don't calculate an integrity check at all. This is usually a bad idea. This can be useful when integrity of the data is verified by other means anyway.

crc32 Calculate CRC32 using the polynomial from IEEE-802.3 (Ethernet).

crc64 Calculate CRC64 using the polynomial from ECMA-182. This is the default, since it is slightly better than CRC32 at detecting damaged files and the speed difference is negligible.

sha256 Calculate SHA-256. This is somewhat slower than CRC32 and CRC64.

Integrity of the **.xz** headers is always verified with CRC32. It is not possible to change or disable it.

--ignore-check

Don't verify the integrity check of the compressed data when decompressing. The CRC32 values in the **.xz** headers will still be verified normally.

Do not use this option unless you know what you are doing. Possible reasons to use this option:

- Trying to recover data from a corrupt .xz file.
- Speeding up decompression. This matters mostly with SHA-256 or with files that have compressed extremely well. It's recommended to not use this option for this purpose unless the file integrity is verified externally in some other way.

-0 ... -9

Select a compression preset level. The default is **-6**. If multiple preset levels are specified, the last one takes effect. If a custom filter chain was already specified, setting a compression preset level clears the custom filter chain.

The differences between the presets are more significant than with **gzip(1)** and **bzip2(1)**. The selected compression settings determine the memory requirements of the decompressor, thus using a too high preset level might make it painful to decompress the file on an old system with little RAM. Specifically, **it's not a good idea to blindly use -9 for everything** like it often is with **gzip(1)** and **bzip2(1)**.

-0 ... -3

These are somewhat fast presets. **-0** is sometimes faster than **gzip -9** while compressing much better. The higher ones often have speed comparable to **bzip2(1)** with comparable or better compression ratio, although the results depend a lot on the type of data being compressed.

-4 ... -6

Good to very good compression while keeping decompressor memory usage reasonable even for old systems. **-6** is the default, which is usually a good choice e.g. for distributing files that need to be decompressible even on systems with only 16 MiB RAM. (**-5e** or **-6e** may be worth considering too. See **--extreme**.)

-7 ... -9

These are like **-6** but with higher compressor and decompressor memory requirements. These are useful only when compressing files bigger than 8 MiB, 16 MiB, and 32 MiB, respectively.

On the same hardware, the decompression speed is approximately a constant number of bytes of compressed data per second. In other words, the better the compression, the faster the decompression will usually be. This also means that the amount of uncompressed output produced per second can vary a lot.

The following table summarises the features of the presets:

| Preset | DictSize | CompCPU | CompMem | DecMem |
|--------|----------|---------|---------|--------|
| -0 | 256 KiB | 0 | 3 MiB | 1 MiB |
| -1 | 1 MiB | 1 | 9 MiB | 2 MiB |
| -2 | 2 MiB | 2 | 17 MiB | 3 MiB |
| -3 | 4 MiB | 3 | 32 MiB | 5 MiB |
| -4 | 4 MiB | 4 | 48 MiB | 5 MiB |
| -5 | 8 MiB | 5 | 94 MiB | 9 MiB |
| -6 | 8 MiB | 6 | 94 MiB | 9 MiB |
| -7 | 16 MiB | 6 | 186 MiB | 17 MiB |
| -8 | 32 MiB | 6 | 370 MiB | 33 MiB |
| -9 | 64 MiB | 6 | 674 MiB | 65 MiB |

Column descriptions:

- DictSize is the LZMA2 dictionary size. It is waste of memory to use a dictionary bigger than the size of the uncompressed file. This is why it is good to avoid using the presets **-7 ... -9** when there's no real need for them. At **-6** and lower, the amount of memory wasted is usually low enough to not matter.
- CompCPU is a simplified representation of the LZMA2 settings that affect compression speed. The dictionary size affects speed too, so while CompCPU is the same for levels **-6 ... -9**,

higher levels still tend to be a little slower. To get even slower and thus possibly better compression, see **--extreme**.

- CompMem contains the compressor memory requirements in the single-threaded mode. It may vary slightly between **xz** versions. Memory requirements of some of the future multi-threaded modes may be dramatically higher than that of the single-threaded mode.
- DecMem contains the decompressor memory requirements. That is, the compression settings determine the memory requirements of the decompressor. The exact decompressor memory usage is slightly more than the LZMA2 dictionary size, but the values in the table have been rounded up to the next full MiB.

-e, --extreme

Use a slower variant of the selected compression preset level (**-0 ... -9**) to hopefully get a little bit better compression ratio, but with bad luck this can also make it worse. Decompressor memory usage is not affected, but compressor memory usage increases a little at preset levels **-0 ... -3**.

Since there are two presets with dictionary sizes 4 MiB and 8 MiB, the presets **-3e** and **-5e** use slightly faster settings (lower CompCPU) than **-4e** and **-6e**, respectively. That way no two presets are identical.

| Preset | DictSize | CompCPU | CompMem | DecMem |
|------------|----------|---------|---------|--------|
| -0e | 256 KiB | 8 | 4 MiB | 1 MiB |
| -1e | 1 MiB | 8 | 13 MiB | 2 MiB |
| -2e | 2 MiB | 8 | 25 MiB | 3 MiB |
| -3e | 4 MiB | 7 | 48 MiB | 5 MiB |
| -4e | 4 MiB | 8 | 48 MiB | 5 MiB |
| -5e | 8 MiB | 7 | 94 MiB | 9 MiB |
| -6e | 8 MiB | 8 | 94 MiB | 9 MiB |
| -7e | 16 MiB | 8 | 186 MiB | 17 MiB |
| -8e | 32 MiB | 8 | 370 MiB | 33 MiB |
| -9e | 64 MiB | 8 | 674 MiB | 65 MiB |

For example, there are a total of four presets that use 8 MiB dictionary, whose order from the fastest to the slowest is **-5**, **-6**, **-5e**, and **-6e**.

--fast

--best These are somewhat misleading aliases for **-0** and **-9**, respectively. These are provided only for backwards compatibility with LZMA Utils. Avoid using these options.

--block-size=size

When compressing to the **.xz** format, split the input data into blocks of *size* bytes. The blocks are compressed independently from each other, which helps with multi-threading and makes limited random-access decompression possible. This option is typically used to override the default block size in multi-threaded mode, but this option can be used in single-threaded mode too.

In multi-threaded mode about three times *size* bytes will be allocated in each thread for buffering input and output. The default *size* is three times the LZMA2 dictionary size or 1 MiB, whichever is more. Typically a good value is 2–4 times the size of the LZMA2 dictionary or at least 1 MiB. Using *size* less than the LZMA2 dictionary size is waste of RAM because then the LZMA2 dictionary buffer will never get fully used. The sizes of the blocks are stored in the block headers, which a future version of **xz** will use for multi-threaded decompression.

In single-threaded mode no block splitting is done by default. Setting this option doesn't affect memory usage. No size information is stored in block headers, thus files created in single-threaded mode won't be identical to files created in multi-threaded mode. The lack of size information also means that a future version of **xz** won't be able to decompress the files in multi-threaded mode.

--block-list=sizes

When compressing to the **.xz** format, start a new block after the given intervals of uncompressed data.

The uncompressed *sizes* of the blocks are specified as a comma-separated list. Omitting a size (two or more consecutive commas) is a shorthand to use the size of the previous block.

If the input file is bigger than the sum of *sizes*, the last value in *sizes* is repeated until the end of the file. A special value of **0** may be used as the last value to indicate that the rest of the file should be encoded as a single block.

If one specifies *sizes* that exceed the encoder's block size (either the default value in threaded mode or the value specified with **--block-size=size**), the encoder will create additional blocks while keeping the boundaries specified in *sizes*. For example, if one specifies **--block-size=10MiB --block-list=5MiB,10MiB,8MiB,12MiB,24MiB** and the input file is 80 MiB, one will get 11 blocks: 5, 10, 8, 10, 2, 10, 10, 4, 10, 10, and 1 MiB.

In multi-threaded mode the sizes of the blocks are stored in the block headers. This isn't done in single-threaded mode, so the encoded output won't be identical to that of the multi-threaded mode.

--flush-timeout=timeout

When compressing, if more than *timeout* milliseconds (a positive integer) has passed since the previous flush and reading more input would block, all the pending input data is flushed from the encoder and made available in the output stream. This can be useful if **xz** is used to compress data that is streamed over a network. Small *timeout* values make the data available at the receiving end with a small delay, but large *timeout* values give better compression ratio.

This feature is disabled by default. If this option is specified more than once, the last one takes effect. The special *timeout* value of **0** can be used to explicitly disable this feature.

This feature is not available on non-POSIX systems.

This feature is still experimental. Currently **xz** is unsuitable for decompressing the stream in real time due to how **xz** does buffering.

--memlimit-compress=limit

Set a memory usage limit for compression. If this option is specified multiple times, the last one takes effect.

If the compression settings exceed the *limit*, **xz** will adjust the settings downwards so that the limit is no longer exceeded and display a notice that automatic adjustment was done. Such adjustments are not made when compressing with **--format=raw** or if **--no-adjust** has been specified. In those cases, an error is displayed and **xz** will exit with exit status 1.

The *limit* can be specified in multiple ways:

- The *limit* can be an absolute value in bytes. Using an integer suffix like **MiB** can be useful. Example: **--memlimit-compress=80MiB**
- The *limit* can be specified as a percentage of total physical memory (RAM). This can be useful especially when setting the **XZ_DEFAULTS** environment variable in a shell initialization script that is shared between different computers. That way the limit is automatically bigger on systems with more memory. Example: **--memlimit-compress=70%**
- The *limit* can be reset back to its default value by setting it to **0**. This is currently equivalent to setting the *limit* to **max** (no memory usage limit). Once multithreading support has been implemented, there may be a difference between **0** and **max** for the multithreaded case, so it is recommended to use **0** instead of **max** until the details have been decided.

See also the section **Memory usage**.

--memlimit-decompress=limit

Set a memory usage limit for decompression. This also affects the **--list** mode. If the operation is not possible without exceeding the *limit*, **xz** will display an error and decompressing the file will

fail. See `--memlimit-compress=limit` for possible ways to specify the *limit*.

-M *limit*, `--memlimit=limit`, `--memory=limit`

This is equivalent to specifying `--memlimit-compress=limit --memlimit-decompress=limit`.

`--no-adjust`

Display an error and exit if the compression settings exceed the memory usage limit. The default is to adjust the settings downwards so that the memory usage limit is not exceeded. Automatic adjusting is always disabled when creating raw streams (`--format=raw`).

-T *threads*, `--threads=threads`

Specify the number of worker threads to use. Setting *threads* to a special value **0** makes **xz** use as many threads as there are CPU cores on the system. The actual number of threads can be less than *threads* if the input file is not big enough for threading with the given settings or if using more threads would exceed the memory usage limit.

Currently the only threading method is to split the input into blocks and compress them independently from each other. The default block size depends on the compression level and can be overridden with the `--block-size=size` option.

Threaded decompression hasn't been implemented yet. It will only work on files that contain multiple blocks with size information in block headers. All files compressed in multi-threaded mode meet this condition, but files compressed in single-threaded mode don't even if `--block-size=size` is used.

Custom compressor filter chains

A custom filter chain allows specifying the compression settings in detail instead of relying on the settings associated to the presets. When a custom filter chain is specified, preset options (`-0 ... -9` and `--extreme`) earlier on the command line are forgotten. If a preset option is specified after one or more custom filter chain options, the new preset takes effect and the custom filter chain options specified earlier are forgotten.

A filter chain is comparable to piping on the command line. When compressing, the uncompressed input goes to the first filter, whose output goes to the next filter (if any). The output of the last filter gets written to the compressed file. The maximum number of filters in the chain is four, but typically a filter chain has only one or two filters.

Many filters have limitations on where they can be in the filter chain: some filters can work only as the last filter in the chain, some only as a non-last filter, and some work in any position in the chain. Depending on the filter, this limitation is either inherent to the filter design or exists to prevent security issues.

A custom filter chain is specified by using one or more filter options in the order they are wanted in the filter chain. That is, the order of filter options is significant! When decoding raw streams (`--format=raw`), the filter chain is specified in the same order as it was specified when compressing.

Filters take filter-specific *options* as a comma-separated list. Extra commas in *options* are ignored. Every option has a default value, so you need to specify only those you want to change.

To see the whole filter chain and *options*, use **xz -vv** (that is, use `--verbose` twice). This works also for viewing the filter chain options used by presets.

`--lzma1[=options]`

`--lzma2[=options]`

Add LZMA1 or LZMA2 filter to the filter chain. These filters can be used only as the last filter in the chain.

LZMA1 is a legacy filter, which is supported almost solely due to the legacy **.lzma** file format, which supports only LZMA1. LZMA2 is an updated version of LZMA1 to fix some practical issues of LZMA1. The **.xz** format uses LZMA2 and doesn't support LZMA1 at all. Compression speed and ratios of LZMA1 and LZMA2 are practically the same.

LZMA1 and LZMA2 share the same set of *options*:

preset=*preset*

Reset all LZMA1 or LZMA2 *options* to *preset*. *Preset* consist of an integer, which may be followed by single-letter preset modifiers. The integer can be from **0** to **9**, matching the command line options **-0** ... **-9**. The only supported modifier is currently **e**, which matches **--extreme**. If no **preset** is specified, the default values of LZMA1 or LZMA2 *options* are taken from the preset **6**.

dict=*size*

Dictionary (history buffer) *size* indicates how many bytes of the recently processed uncompressed data is kept in memory. The algorithm tries to find repeating byte sequences (matches) in the uncompressed data, and replace them with references to the data currently in the dictionary. The bigger the dictionary, the higher is the chance to find a match. Thus, increasing dictionary *size* usually improves compression ratio, but a dictionary bigger than the uncompressed file is waste of memory.

Typical dictionary *size* is from 64 KiB to 64 MiB. The minimum is 4 KiB. The maximum for compression is currently 1.5 GiB (1536 MiB). The decompressor already supports dictionaries up to one byte less than 4 GiB, which is the maximum for the LZMA1 and LZMA2 stream formats.

Dictionary *size* and match finder (*mf*) together determine the memory usage of the LZMA1 or LZMA2 encoder. The same (or bigger) dictionary *size* is required for decompressing that was used when compressing, thus the memory usage of the decoder is determined by the dictionary size used when compressing. The **.xz** headers store the dictionary *size* either as 2^n or $2^n + 2^{(n-1)}$, so these *sizes* are somewhat preferred for compression. Other *sizes* will get rounded up when stored in the **.xz** headers.

lc=*lc* Specify the number of literal context bits. The minimum is 0 and the maximum is 4; the default is 3. In addition, the sum of *lc* and *lp* must not exceed 4.

All bytes that cannot be encoded as matches are encoded as literals. That is, literals are simply 8-bit bytes that are encoded one at a time.

The literal coding makes an assumption that the highest *lc* bits of the previous uncompressed byte correlate with the next byte. E.g. in typical English text, an upper-case letter is often followed by a lower-case letter, and a lower-case letter is usually followed by another lower-case letter. In the US-ASCII character set, the highest three bits are 010 for upper-case letters and 011 for lower-case letters. When *lc* is at least 3, the literal coding can take advantage of this property in the uncompressed data.

The default value (3) is usually good. If you want maximum compression, test **lc=4**. Sometimes it helps a little, and sometimes it makes compression worse. If it makes it worse, test e.g. **lc=2** too.

lp=*lp* Specify the number of literal position bits. The minimum is 0 and the maximum is 4; the default is 0.

Lp affects what kind of alignment in the uncompressed data is assumed when encoding literals. See *pb* below for more information about alignment.

pb=*pb* Specify the number of position bits. The minimum is 0 and the maximum is 4; the default is 2.

Pb affects what kind of alignment in the uncompressed data is assumed in general. The default means four-byte alignment ($2^{pb}=2^2=4$), which is often a good choice when there's no better guess.

When the alignment is known, setting *pb* accordingly may reduce the file size a little. E.g. with text files having one-byte alignment (US-ASCII, ISO-8859-*, UTF-8), setting **pb=0** can improve compression slightly. For UTF-16 text, **pb=1** is a good choice. If the alignment is an odd number like 3 bytes, **pb=0** might be the best choice.

Even though the assumed alignment can be adjusted with *pb* and *lp*, LZMA1 and LZMA2 still slightly favor 16-byte alignment. It might be worth taking into account when designing file formats that are likely to be often compressed with LZMA1 or LZMA2.

mf=mf Match finder has a major effect on encoder speed, memory usage, and compression ratio. Usually Hash Chain match finders are faster than Binary Tree match finders. The default depends on the *preset*: 0 uses **hc3**, 1–3 use **hc4**, and the rest use **bt4**.

The following match finders are supported. The memory usage formulas below are rough approximations, which are closest to the reality when *dict* is a power of two.

- hc3** Hash Chain with 2- and 3-byte hashing
Minimum value for *nice*: 3
Memory usage:
dict * 7.5 (if *dict* ≤ 16 MiB);
dict * 5.5 + 64 MiB (if *dict* > 16 MiB)
- hc4** Hash Chain with 2-, 3-, and 4-byte hashing
Minimum value for *nice*: 4
Memory usage:
dict * 7.5 (if *dict* ≤ 32 MiB);
dict * 6.5 (if *dict* > 32 MiB)
- bt2** Binary Tree with 2-byte hashing
Minimum value for *nice*: 2
Memory usage: *dict* * 9.5
- bt3** Binary Tree with 2- and 3-byte hashing
Minimum value for *nice*: 3
Memory usage:
dict * 11.5 (if *dict* ≤ 16 MiB);
dict * 9.5 + 64 MiB (if *dict* > 16 MiB)
- bt4** Binary Tree with 2-, 3-, and 4-byte hashing
Minimum value for *nice*: 4
Memory usage:
dict * 11.5 (if *dict* ≤ 32 MiB);
dict * 10.5 (if *dict* > 32 MiB)

mode=mode

Compression *mode* specifies the method to analyze the data produced by the match finder. Supported *modes* are **fast** and **normal**. The default is **fast** for *presets* 0–3 and **normal** for *presets* 4–9.

Usually **fast** is used with Hash Chain match finders and **normal** with Binary Tree match finders. This is also what the *presets* do.

nice=nice

Specify what is considered to be a nice length for a match. Once a match of at least *nice* bytes is found, the algorithm stops looking for possibly better matches.

Nice can be 2–273 bytes. Higher values tend to give better compression ratio at the expense of speed. The default depends on the *preset*.

depth=depth

Specify the maximum search depth in the match finder. The default is the special value of 0, which makes the compressor determine a reasonable *depth* from *mf* and *nice*.

Reasonable *depth* for Hash Chains is 4–100 and 16–1000 for Binary Trees. Using very high values for *depth* can make the encoder extremely slow with some files. Avoid setting the *depth* over 1000 unless you are prepared to interrupt the compression in case it is

taking far too long.

When decoding raw streams (**--format=raw**), LZMA2 needs only the dictionary *size*. LZMA1 needs also *lc*, *lp*, and *pb*.

--x86[=*options*]
--powerpc[=*options*]
--ia64[=*options*]
--arm[=*options*]
--armthumb[=*options*]
--sparc[=*options*]

Add a branch/call/jump (BCJ) filter to the filter chain. These filters can be used only as a non-last filter in the filter chain.

A BCJ filter converts relative addresses in the machine code to their absolute counterparts. This doesn't change the size of the data, but it increases redundancy, which can help LZMA2 to produce 0–15 % smaller **.xz** file. The BCJ filters are always reversible, so using a BCJ filter for wrong type of data doesn't cause any data loss, although it may make the compression ratio slightly worse.

It is fine to apply a BCJ filter on a whole executable; there's no need to apply it only on the executable section. Applying a BCJ filter on an archive that contains both executable and non-executable files may or may not give good results, so it generally isn't good to blindly apply a BCJ filter when compressing binary packages for distribution.

These BCJ filters are very fast and use insignificant amount of memory. If a BCJ filter improves compression ratio of a file, it can improve decompression speed at the same time. This is because, on the same hardware, the decompression speed of LZMA2 is roughly a fixed number of bytes of compressed data per second.

These BCJ filters have known problems related to the compression ratio:

- Some types of files containing executable code (e.g. object files, static libraries, and Linux kernel modules) have the addresses in the instructions filled with filler values. These BCJ filters will still do the address conversion, which will make the compression worse with these files.
- Applying a BCJ filter on an archive containing multiple similar executables can make the compression ratio worse than not using a BCJ filter. This is because the BCJ filter doesn't detect the boundaries of the executable files, and doesn't reset the address conversion counter for each executable.

Both of the above problems will be fixed in the future in a new filter. The old BCJ filters will still be useful in embedded systems, because the decoder of the new filter will be bigger and use more memory.

Different instruction sets have have different alignment:

| Filter | Alignment | Notes |
|-----------|-----------|----------------------|
| x86 | 1 | 32-bit or 64-bit x86 |
| PowerPC | 4 | Big endian only |
| ARM | 4 | Little endian only |
| ARM-Thumb | 2 | Little endian only |
| IA-64 | 16 | Big or little endian |
| SPARC | 4 | Big or little endian |

Since the BCJ-filtered data is usually compressed with LZMA2, the compression ratio may be improved slightly if the LZMA2 options are set to match the alignment of the selected BCJ filter. For example, with the IA-64 filter, it's good to set **pb=4** with LZMA2 ($2^4=16$). The x86 filter is an exception; it's usually good to stick to LZMA2's default four-byte alignment when compressing x86 executables.

All BCJ filters support the same *options*:

start=offset

Specify the start *offset* that is used when converting between relative and absolute addresses. The *offset* must be a multiple of the alignment of the filter (see the table above). The default is zero. In practice, the default is good; specifying a custom *offset* is almost never useful.

--delta[=options]

Add the Delta filter to the filter chain. The Delta filter can be only used as a non-last filter in the filter chain.

Currently only simple byte-wise delta calculation is supported. It can be useful when compressing e.g. uncompressed bitmap images or uncompressed PCM audio. However, special purpose algorithms may give significantly better results than Delta + LZMA2. This is true especially with audio, which compresses faster and better e.g. with **flac**(1).

Supported *options*:

dist=distance

Specify the *distance* of the delta calculation in bytes. *distance* must be 1–256. The default is 1.

For example, with **dist=2** and eight-byte input A1 B1 A2 B3 A3 B5 A4 B7, the output will be A1 B1 01 02 01 02 01 02.

Other options

-q, --quiet

Suppress warnings and notices. Specify this twice to suppress errors too. This option has no effect on the exit status. That is, even if a warning was suppressed, the exit status to indicate a warning is still used.

-v, --verbose

Be verbose. If standard error is connected to a terminal, **xz** will display a progress indicator. Specifying **--verbose** twice will give even more verbose output.

The progress indicator shows the following information:

- Completion percentage is shown if the size of the input file is known. That is, the percentage cannot be shown in pipes.
- Amount of compressed data produced (compressing) or consumed (decompressing).
- Amount of uncompressed data consumed (compressing) or produced (decompressing).
- Compression ratio, which is calculated by dividing the amount of compressed data processed so far by the amount of uncompressed data processed so far.
- Compression or decompression speed. This is measured as the amount of uncompressed data consumed (compression) or produced (decompression) per second. It is shown after a few seconds have passed since **xz** started processing the file.
- Elapsed time in the format M:SS or H:MM:SS.
- Estimated remaining time is shown only when the size of the input file is known and a couple of seconds have already passed since **xz** started processing the file. The time is shown in a less precise format which never has any colons, e.g. 2 min 30 s.

When standard error is not a terminal, **--verbose** will make **xz** print the filename, compressed size, uncompressed size, compression ratio, and possibly also the speed and elapsed time on a single line to standard error after compressing or decompressing the file. The speed and elapsed time are included only when the operation took at least a few seconds. If the operation didn't finish, e.g. due to user interruption, also the completion percentage is printed if the size of the input file is known.

-Q, --no-warn

Don't set the exit status to 2 even if a condition worth a warning was detected. This option doesn't affect the verbosity level, thus both **--quiet** and **--no-warn** have to be used to not display warnings and to not alter the exit status.

--robot

Print messages in a machine-parsable format. This is intended to ease writing frontends that want to use **xz** instead of liblzma, which may be the case with various scripts. The output with this option enabled is meant to be stable across **xz** releases. See the section **ROBOT MODE** for details.

--info-memory

Display, in human-readable format, how much physical memory (RAM) **xz** thinks the system has and the memory usage limits for compression and decompression, and exit successfully.

-h, --help

Display a help message describing the most commonly used options, and exit successfully.

-H, --long-help

Display a help message describing all features of **xz**, and exit successfully

-V, --version

Display the version number of **xz** and liblzma in human readable format. To get machine-parsable output, specify **--robot** before **--version**.

ROBOT MODE

The robot mode is activated with the **--robot** option. It makes the output of **xz** easier to parse by other programs. Currently **--robot** is supported only together with **--version**, **--info-memory**, and **--list**. It will be supported for compression and decompression in the future.

Version

xz --robot --version will print the version number of **xz** and liblzma in the following format:

XZ_VERSION=YYYYZZZS

LIBLZMA_VERSION=YYYYZZZS

X Major version.

YYY Minor version. Even numbers are stable. Odd numbers are alpha or beta versions.

ZZZ Patch level for stable releases or just a counter for development releases.

S Stability. 0 is alpha, 1 is beta, and 2 is stable. **S** should be always 2 when **YYY** is even.

YYYYZZZS are the same on both lines if **xz** and liblzma are from the same XZ Utils release.

Examples: 4.999.9beta is **49990091** and 5.0.0 is **50000002**.

Memory limit information

xz --robot --info-memory prints a single line with three tab-separated columns:

1. Total amount of physical memory (RAM) in bytes
2. Memory usage limit for compression in bytes. A special value of zero indicates the default setting, which for single-threaded mode is the same as no limit.
3. Memory usage limit for decompression in bytes. A special value of zero indicates the default setting, which for single-threaded mode is the same as no limit.

In the future, the output of **xz --robot --info-memory** may have more columns, but never more than a single line.

List mode

xz --robot --list uses tab-separated output. The first column of every line has a string that indicates the type of the information found on that line:

- name** This is always the first line when starting to list a file. The second column on the line is the file-name.
- file** This line contains overall information about the **.xz** file. This line is always printed after the **name** line.
- stream** This line type is used only when **--verbose** was specified. There are as many **stream** lines as there are streams in the **.xz** file.
- block** This line type is used only when **--verbose** was specified. There are as many **block** lines as there are blocks in the **.xz** file. The **block** lines are shown after all the **stream** lines; different line types are not interleaved.

summary

This line type is used only when **--verbose** was specified twice. This line is printed after all **block** lines. Like the **file** line, the **summary** line contains overall information about the **.xz** file.

- totals** This line is always the very last line of the list output. It shows the total counts and sizes.

The columns of the **file** lines:

2. Number of streams in the file
3. Total number of blocks in the stream(s)
4. Compressed size of the file
5. Uncompressed size of the file
6. Compression ratio, for example **0.123**. If ratio is over 9.999, three dashes (---) are displayed instead of the ratio.
7. Comma-separated list of integrity check names. The following strings are used for the known check types: **None**, **CRC32**, **CRC64**, and **SHA-256**. For unknown check types, **Unknown-N** is used, where *N* is the Check ID as a decimal number (one or two digits).
8. Total size of stream padding in the file

The columns of the **stream** lines:

2. Stream number (the first stream is 1)
3. Number of blocks in the stream
4. Compressed start offset
5. Uncompressed start offset
6. Compressed size (does not include stream padding)
7. Uncompressed size
8. Compression ratio
9. Name of the integrity check
10. Size of stream padding

The columns of the **block** lines:

2. Number of the stream containing this block
3. Block number relative to the beginning of the stream (the first block is 1)
4. Block number relative to the beginning of the file
5. Compressed start offset relative to the beginning of the file
6. Uncompressed start offset relative to the beginning of the file
7. Total compressed size of the block (includes headers)
8. Uncompressed size
9. Compression ratio
10. Name of the integrity check

If **--verbose** was specified twice, additional columns are included on the **block** lines. These are not displayed with a single **--verbose**, because getting this information requires many seeks and can thus be slow:

11. Value of the integrity check in hexadecimal
12. Block header size
13. Block flags: **c** indicates that compressed size is present, and **u** indicates that uncompressed size is present. If the flag is not set, a dash (-) is shown instead to keep the string length

- fixed. New flags may be added to the end of the string in the future.
14. Size of the actual compressed data in the block (this excludes the block header, block padding, and check fields)
 15. Amount of memory (in bytes) required to decompress this block with this **xz** version
 16. Filter chain. Note that most of the options used at compression time cannot be known, because only the options that are needed for decompression are stored in the **.xz** headers.

The columns of the **summary** lines:

2. Amount of memory (in bytes) required to decompress this file with this **xz** version
3. **yes** or **no** indicating if all block headers have both compressed size and uncompressed size stored in them

Since **xz 5.1.2alpha**:

4. Minimum **xz** version required to decompress the file

The columns of the **totals** line:

2. Number of streams
3. Number of blocks
4. Compressed size
5. Uncompressed size
6. Average compression ratio
7. Comma-separated list of integrity check names that were present in the files
8. Stream padding size
9. Number of files. This is here to keep the order of the earlier columns the same as on **file** lines.

If **--verbose** was specified twice, additional columns are included on the **totals** line:

10. Maximum amount of memory (in bytes) required to decompress the files with this **xz** version
11. **yes** or **no** indicating if all block headers have both compressed size and uncompressed size stored in them

Since **xz 5.1.2alpha**:

12. Minimum **xz** version required to decompress the file

Future versions may add new line types and new columns can be added to the existing line types, but the existing columns won't be changed.

EXIT STATUS

- | | |
|----------|--|
| 0 | All is good. |
| 1 | An error occurred. |
| 2 | Something worth a warning occurred, but no actual errors occurred. |

Notices (not warnings or errors) printed on standard error don't affect the exit status.

ENVIRONMENT

xz parses space-separated lists of options from the environment variables **XZ_DEFAULTS** and **XZ_OPT**, in this order, before parsing the options from the command line. Note that only options are parsed from the environment variables; all non-options are silently ignored. Parsing is done with **getopt_long(3)** which is used also for the command line arguments.

XZ_DEFAULTS

User-specific or system-wide default options. Typically this is set in a shell initialization script to enable **xz**'s memory usage limiter by default. Excluding shell initialization scripts and similar special cases, scripts must never set or unset **XZ_DEFAULTS**.

XZ_OPT

This is for passing options to **xz** when it is not possible to set the options directly on the **xz** command line. This is the case e.g. when **xz** is run by a script or tool, e.g. GNU **tar(1)**:

```
XZ_OPT=-2v tar caf foo.tar.xz foo
```

Scripts may use **XZ_OPT** e.g. to set script-specific default compression options. It is still recommended to allow users to override **XZ_OPT** if that is reasonable, e.g. in **sh**(1) scripts one may use something like this:

```
XZ_OPT=${XZ_OPT:-"-7e"}
export XZ_OPT
```

LZMA UTILS COMPATIBILITY

The command line syntax of **xz** is practically a superset of **lzma**, **unlzma**, and **lzcat** as found from LZMA Utils 4.32.x. In most cases, it is possible to replace LZMA Utils with XZ Utils without breaking existing scripts. There are some incompatibilities though, which may sometimes cause problems.

Compression preset levels

The numbering of the compression level presets is not identical in **xz** and LZMA Utils. The most important difference is how dictionary sizes are mapped to different presets. Dictionary size is roughly equal to the decompressor memory usage.

| Level | xz | LZMA Utils |
|-------|---------|------------|
| -0 | 256 KiB | N/A |
| -1 | 1 MiB | 64 KiB |
| -2 | 2 MiB | 1 MiB |
| -3 | 4 MiB | 512 KiB |
| -4 | 4 MiB | 1 MiB |
| -5 | 8 MiB | 2 MiB |
| -6 | 8 MiB | 4 MiB |
| -7 | 16 MiB | 8 MiB |
| -8 | 32 MiB | 16 MiB |
| -9 | 64 MiB | 32 MiB |

The dictionary size differences affect the compressor memory usage too, but there are some other differences between LZMA Utils and XZ Utils, which make the difference even bigger:

| Level | xz | LZMA Utils 4.32.x |
|-------|---------|-------------------|
| -0 | 3 MiB | N/A |
| -1 | 9 MiB | 2 MiB |
| -2 | 17 MiB | 12 MiB |
| -3 | 32 MiB | 12 MiB |
| -4 | 48 MiB | 16 MiB |
| -5 | 94 MiB | 26 MiB |
| -6 | 94 MiB | 45 MiB |
| -7 | 186 MiB | 83 MiB |
| -8 | 370 MiB | 159 MiB |
| -9 | 674 MiB | 311 MiB |

The default preset level in LZMA Utils is **-7** while in XZ Utils it is **-6**, so both use an 8 MiB dictionary by default.

Streamed vs. non-streamed .lzma files

The uncompressed size of the file can be stored in the **.lzma** header. LZMA Utils does that when compressing regular files. The alternative is to mark that uncompressed size is unknown and use end-of-payload marker to indicate where the decompressor should stop. LZMA Utils uses this method when uncompressed size isn't known, which is the case for example in pipes.

xz supports decompressing **.lzma** files with or without end-of-payload marker, but all **.lzma** files created by **xz** will use end-of-payload marker and have uncompressed size marked as unknown in the **.lzma** header. This may be a problem in some uncommon situations. For example, a **.lzma** decompressor in an embedded device might work only with files that have known uncompressed size. If you hit this problem, you need to use LZMA Utils or LZMA SDK to create **.lzma** files with known uncompressed size.

Unsupported .lzma files

The **.lzma** format allows *lc* values up to 8, and *lp* values up to 4. LZMA Utils can decompress files with any *lc* and *lp*, but always creates files with **lc=3** and **lp=0**. Creating files with other *lc* and *lp* is possible with **xz** and with LZMA SDK.

The implementation of the LZMA1 filter in liblzma requires that the sum of *lc* and *lp* must not exceed 4. Thus, **.lzma** files, which exceed this limitation, cannot be decompressed with **xz**.

LZMA Utils creates only **.lzma** files which have a dictionary size of 2^n (a power of 2) but accepts files with any dictionary size. liblzma accepts only **.lzma** files which have a dictionary size of 2^n or $2^n + 2^{(n-1)}$. This is to decrease false positives when detecting **.lzma** files.

These limitations shouldn't be a problem in practice, since practically all **.lzma** files have been compressed with settings that liblzma will accept.

Trailing garbage

When decompressing, LZMA Utils silently ignore everything after the first **.lzma** stream. In most situations, this is a bug. This also means that LZMA Utils don't support decompressing concatenated **.lzma** files.

If there is data left after the first **.lzma** stream, **xz** considers the file to be corrupt unless **--single-stream** was used. This may break obscure scripts which have assumed that trailing garbage is ignored.

NOTES

Compressed output may vary

The exact compressed output produced from the same uncompressed input file may vary between XZ Utils versions even if compression options are identical. This is because the encoder can be improved (faster or better compression) without affecting the file format. The output can vary even between different builds of the same XZ Utils version, if different build options are used.

The above means that once **--rsyncable** has been implemented, the resulting files won't necessarily be rsyncable unless both old and new files have been compressed with the same xz version. This problem can be fixed if a part of the encoder implementation is frozen to keep rsyncable output stable across xz versions.

Embedded .xz decompressors

Embedded **.xz** decompressor implementations like XZ Embedded don't necessarily support files created with integrity *check* types other than **none** and **crc32**. Since the default is **--check=crc64**, you must use **--check=none** or **--check=crc32** when creating files for embedded systems.

Outside embedded systems, all **.xz** format decompressors support all the *check* types, or at least are able to decompress the file without verifying the integrity check if the particular *check* is not supported.

XZ Embedded supports BCJ filters, but only with the default start offset.

EXAMPLES

Basics

Compress the file *foo* into *foo.xz* using the default compression level (**-6**), and remove *foo* if compression is successful:

```
xz foo
```

Decompress *bar.xz* into *bar* and don't remove *bar.xz* even if decompression is successful:

```
xz -dk bar.xz
```

Create *baz.tar.xz* with the preset **-4e** (**-4 --extreme**), which is slower than e.g. the default **-6**, but needs less memory for compression and decompression (48 MiB and 5 MiB, respectively):

```
tar cf - baz | xz -4e > baz.tar.xz
```

A mix of compressed and uncompressed files can be decompressed to standard output with a single command:

```
xz -dcf a.txt b.txt.xz c.txt d.txt.lzma > abcd.txt
```

Parallel compression of many files

On GNU and *BSD, **find**(1) and **xargs**(1) can be used to parallelize compression of many files:

```
find . -type f \! -name '*.xz' -print0 \
| xargs -0r -P4 -n16 xz -T1
```

The **-P** option to **xargs**(1) sets the number of parallel **xz** processes. The best value for the **-n** option depends on how many files there are to be compressed. If there are only a couple of files, the value should probably be 1; with tens of thousands of files, 100 or even more may be appropriate to reduce the number of **xz** processes that **xargs**(1) will eventually create.

The option **-T1** for **xz** is there to force it to single-threaded mode, because **xargs**(1) is used to control the amount of parallelization.

Robot mode

Calculate how many bytes have been saved in total after compressing multiple files:

```
xz --robot --list *.xz | awk '/^totals/{print $5-$4}'
```

A script may want to know that it is using new enough **xz**. The following **sh**(1) script checks that the version number of the **xz** tool is at least 5.0.0. This method is compatible with old beta versions, which didn't support the **--robot** option:

```
if ! eval "$(xz --robot --version 2> /dev/null)" ||
[ "$XZ_VERSION" -lt 50000002 ]; then
    echo "Your xz is too old."
fi
unset XZ_VERSION LIBLZMA_VERSION
```

Set a memory usage limit for decompression using **XZ_OPT**, but if a limit has already been set, don't increase it:

```
NEWLIM=$((123 << 20)) # 123 MiB
OLDLIM=$(xz --robot --info-memory | cut -f3)
if [ $OLDLIM -eq 0 -o $OLDLIM -gt $NEWLIM ]; then
    XZ_OPT="$XZ_OPT --memlimit-decompress=$NEWLIM"
    export XZ_OPT
fi
```

Custom compressor filter chains

The simplest use for custom filter chains is customizing a LZMA2 preset. This can be useful, because the presets cover only a subset of the potentially useful combinations of compression settings.

The CompCPU columns of the tables from the descriptions of the options **-0 ... -9** and **--extreme** are useful when customizing LZMA2 presets. Here are the relevant parts collected from those two tables:

| Preset | CompCPU |
|--------|---------|
| -0 | 0 |
| -1 | 1 |
| -2 | 2 |
| -3 | 3 |
| -4 | 4 |
| -5 | 5 |
| -6 | 6 |
| -5e | 7 |
| -6e | 8 |

If you know that a file requires somewhat big dictionary (e.g. 32 MiB) to compress well, but you want to compress it quicker than **xz -8** would do, a preset with a low CompCPU value (e.g. 1) can be modified to use a bigger dictionary:

```
xz --lzma2=preset=1,dict=32MiB foo.tar
```

With certain files, the above command may be faster than **xz -6** while compressing significantly better. However, it must be emphasized that only some files benefit from a big dictionary while keeping the ComCPU value low. The most obvious situation, where a big dictionary can help a lot, is an archive containing very similar files of at least a few megabytes each. The dictionary size has to be significantly bigger than any individual file to allow LZMA2 to take full advantage of the similarities between consecutive files.

If very high compressor and decompressor memory usage is fine, and the file being compressed is at least several hundred megabytes, it may be useful to use an even bigger dictionary than the 64 MiB that **xz -9** would use:

```
xz -vv --lzma2=dict=192MiB big_foo.tar
```

Using **-vv** (**--verbose --verbose**) like in the above example can be useful to see the memory requirements of the compressor and decompressor. Remember that using a dictionary bigger than the size of the uncompressed file is waste of memory, so the above command isn't useful for small files.

Sometimes the compression time doesn't matter, but the decompressor memory usage has to be kept low e.g. to make it possible to decompress the file on an embedded system. The following command uses **-6e** (**-6 --extreme**) as a base and sets the dictionary to only 64 KiB. The resulting file can be decompressed with XZ Embedded (that's why there is **--check=crc32**) using about 100 KiB of memory.

```
xz --check=crc32 --lzma2=preset=6e,dict=64KiB foo
```

If you want to squeeze out as many bytes as possible, adjusting the number of literal context bits (*lc*) and number of position bits (*pb*) can sometimes help. Adjusting the number of literal position bits (*lp*) might help too, but usually *lc* and *pb* are more important. E.g. a source code archive contains mostly US-ASCII text, so something like the following might give slightly (like 0.1 %) smaller file than **xz -6e** (try also without **lc=4**):

```
xz --lzma2=preset=6e,pb=0,lc=4 source_code.tar
```

Using another filter together with LZMA2 can improve compression with certain file types. E.g. to compress a x86-32 or x86-64 shared library using the x86 BCJ filter:

```
xz --x86 --lzma2 libfoo.so
```

Note that the order of the filter options is significant. If **--x86** is specified after **--lzma2**, **xz** will give an error, because there cannot be any filter after LZMA2, and also because the x86 BCJ filter cannot be used as the last filter in the chain.

The Delta filter together with LZMA2 can give good results with bitmap images. It should usually beat PNG, which has a few more advanced filters than simple delta but uses Deflate for the actual compression.

The image has to be saved in uncompressed format, e.g. as uncompressed TIFF. The distance parameter of the Delta filter is set to match the number of bytes per pixel in the image. E.g. 24-bit RGB bitmap needs **dist=3**, and it is also good to pass **pb=0** to LZMA2 to accommodate the three-byte alignment:

```
xz --delta=dist=3 --lzma2=pb=0 foo.tiff
```

If multiple images have been put into a single archive (e.g. **.tar**), the Delta filter will work on that too as long as all images have the same number of bytes per pixel.

SEE ALSO

xzdec(1), **xzdiff(1)**, **xzgrep(1)**, **xzless(1)**, **xzmore(1)**, **gzip(1)**, **bzip2(1)**, **7z(1)**

XZ Utils: <<http://tukaani.org/xz/>>

XZ Embedded: <<http://tukaani.org/xz/embedded.html>>

LZMA SDK: <<http://7-zip.org/sdk.html>>

NAME

yesterday – return the date of yesterday

SYNOPSIS

edrc/lib/yesterday [**-h**]

yesterday ["*date_format*"]

AVAILABILITY

WA2L/edrc

DESCRIPTION

return the date of yesterday. By default the date is returned in the format *YYYY-MM-DD*.

OPTIONS

-h print usage message.

"*date_format*"

date format as known from the **date**(1) command.

The following format place holders are the most compatible ones:

%Y year (1970...)

%y last two digits of year (00..99)

%m month (01..12)

%d day of month (01..31)

%H hour (00..23)

%M minute (00..59)

%S second (00..60)

%w day of week (0..6); 0 represents Sunday

ENVIRONMENT

\$TZ Timezone setting.

EXIT STATUS

0 no error.

yesterday(3)

Library Commands

yesterday(3)

2 date could not be resolved.

4 usage message printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1), **date(1)**, **days(3)**, **is_weekend(3)**, **seconds(3)**, **timer(1)**, **today(3)**, **tomorrow(3)**

NOTES

-

BUGS

-

AUTHOR

yesterday was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2008 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

ypxfr_all – transfer all maps from a NIS master server

SYNOPSIS

edrc/bin/ypxfr_all [-h]

ypxfr_all [-t]

AVAILABILITY

WA2L/edrc

DESCRIPTION

force a transfer of all NIS maps from a NIS master server to the NIS slave server where logged on.

This command is to completely refresh the maps on the NIS slave server independent of the automatic synchronization mechanism.

OPTIONS

- h** usage message.
- t** transfer the following maps from the NIS master server: **auto.apl**, **auto.apl_std**, **auto.dat**, **auto.home**, **auto.master**, **auto.wwr**, **bootparams**, **ethers.byaddr**, **ethers.byname**, **group.bygid**, **group.byname**, **hosts.byaddr**, **hosts.byname**, **mail.aliases**, **mail.byaddr**, **netgroup.byhost**, **netgroup.byuser**, **netgroup**, **netmasks.byaddr**, **networks.byaddr**, **networks.byname**, **passwd.byname**, **passwd.byuid**, **protocols.byname**, **protocols.bynumber**, **restr.byname**, **restr.byuid**, **rpc.bynumber**, **services.byname**, **timezone.byname** and **ypservers**.

ENVIRONMENT

-

EXIT STATUS

- 0** no error.
- 1** the **ypxfr_all** command was not started as user **root** or it was not started on a NIS slave server.
- 4** usage printed.

FILES

-

EXAMPLES

-

SEE ALSO

edrcintro(1)

NOTES

-

BUGS

-

AUTHOR

ypxfr_all was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net.

COPYRIGHT

Copyright © 2020 by Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

NAME

gzip, *gunzip*, *zcat* – compress or expand files

SYNOPSIS

gzip [**-acdfhlLnNrtvV19**] [**-S suffix**] [*name ...*]

gunzip [**-acfhllNrtvV**] [**-S suffix**] [*name ...*]

zcat [**-fhLV**] [*name ...*]

DESCRIPTION

Gzip reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension **.gz**, while keeping the same ownership modes, access and modification times. (The default extension is **-gz** for VMS, **z** for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. *Gzip* will only attempt to compress regular files. In particular, it will ignore symbolic links.

If the compressed file name is too long for its file system, *gzip* truncates it. *Gzip* attempts to truncate only the parts of the file name longer than 3 characters. (A part is delimited by dots.) If the name consists of small parts only, the longest parts are truncated. For example, if file names are limited to 14 characters, *gzip.msdos.exe* is compressed to *gzi.msd.exe.gz*. Names are not truncated on systems which do not have a limit on file name length.

By default, *gzip* keeps the original file name and timestamp in the compressed file. These are used when decompressing the file with the **-N** option. This is useful when the compressed file name was truncated or when the time stamp was not preserved after a file transfer.

Compressed files can be restored to their original form using *gzip -d* or *gunzip* or *zcat*. If the original name saved in the compressed file is not suitable for its file system, a new name is constructed from the original one to make it legal.

gunzip takes a list of files on its command line and replaces each file whose name ends with **.gz**, **-gz**, **.z**, **-z**, **_z** or **.Z** and which begins with the correct magic number with an uncompressed file without the original extension. *gunzip* also recognizes the special extensions **.tgz** and **.taz** as shorthands for **.tar.gz** and **.tar.Z** respectively. When compressing, *gzip* uses the **.tgz** extension if necessary instead of truncating a file with a **.tar** extension.

gunzip can currently decompress files created by *gzip*, *zip*, *compress*, *compress -H* or *pack*. The detection of the input format is automatic. When using the first two formats, *gunzip* checks a 32 bit CRC. For *pack*, *gunzip* checks the uncompressed length. The standard *compress* format was not designed to allow consistency checks. However *gunzip* is sometimes able to detect a bad **.Z** file. If you get an error when uncompressing a **.Z** file, do not assume that the **.Z** file is correct simply because the standard *uncompress* does not complain. This generally means that the standard *uncompress* does not check its input, and happily generates garbage output. The SCO *compress -H* format (lzh compression method) does not include a CRC but also allows some consistency checks.

Files created by *zip* can be uncompressed by *gzip* only if they have a single member compressed with the 'deflation' method. This feature is only intended to help conversion of *tar.zip* files to the *tar.gz* format. To extract a *zip* file with a single member, use a command like *gunzip <foo.zip* or *gunzip -S .zip foo.zip*. To extract *zip* files with several members, use *unzip* instead of *gunzip*.

zcat is identical to *gunzip -c*. (On some systems, *zcat* may be installed as *gzcat* to preserve the original link to *compress*.) *zcat* uncompresses either a list of files on the command line or its standard input and writes the uncompressed data on standard output. *zcat* will uncompress files that have the correct magic number whether they have a **.gz** suffix or not.

Gzip uses the Lempel-Ziv algorithm used in *zip* and PKZIP. The amount of compression obtained depends on the size of the input and the distribution of common substrings. Typically, text such as source code or English is reduced by 60–70%. Compression is generally much better than that achieved by LZW (as used in *compress*), Huffman coding (as used in *pack*), or adaptive Huffman coding (*compact*).

Compression is always performed, even if the compressed file is slightly larger than the original. The worst case expansion is a few bytes for the *gzip* file header, plus 5 bytes every 32K block, or an expansion ratio of

0.015% for large files. Note that the actual number of used disk blocks almost never increases. *gzip* preserves the mode, ownership and timestamps of files when compressing or decompressing.

The *gzip* file format is specified in P. Deutsch, GZIP file format specification version 4.3, <ftp://ftp.isi.edu/in-notes/rfc1952.txt>, Internet RFC 1952 (May 1996). The *zip* deflation format is specified in P. Deutsch, DEFLATE Compressed Data Format Specification version 1.3, <ftp://ftp.isi.edu/in-notes/rfc1951.txt>, Internet RFC 1951 (May 1996).

OPTIONS

-a --ascii

Ascii text mode: convert end-of-lines using local conventions. This option is supported only on some non-Unix systems. For MSDOS, CR LF is converted to LF when compressing, and LF is converted to CR LF when decompressing.

-c --stdout --to-stdout

Write output on standard output; keep original files unchanged. If there are several input files, the output consists of a sequence of independently compressed members. To obtain better compression, concatenate all input files before compressing them.

-d --decompress --uncompress

Decompress.

-f --force

Force compression or decompression even if the file has multiple links or the corresponding file already exists, or if the compressed data is read from or written to a terminal. If the input data is not in a format recognized by *gzip*, and if the option *--stdout* is also given, copy the input data without change to the standard output: let *zcat* behave as *cat*. If *-f* is not given, and when not running in the background, *gzip* prompts to verify whether an existing file should be overwritten.

-h --help

Display a help screen and quit.

-l --list For each compressed file, list the following fields:

compressed size: size of the compressed file
 uncompressed size: size of the uncompressed file
 ratio: compression ratio (0.0% if unknown)
 uncompressed_name: name of the uncompressed file

The uncompressed size is given as -1 for files not in gzip format, such as compressed .Z files. To get the uncompressed size for such a file, you can use:

```
zcat file.Z | wc -c
```

In combination with the *--verbose* option, the following fields are also displayed:

method: compression method
 crc: the 32-bit CRC of the uncompressed data
 date & time: time stamp for the uncompressed file

The compression methods currently supported are deflate, compress, lzh (SCO compress -H) and pack. The crc is given as ffffffff for a file not in gzip format.

With *--name*, the uncompressed name, date and time are those stored within the compress file if present.

With *--verbose*, the size totals and compression ratio for all files is also displayed, unless some sizes are unknown. With *--quiet*, the title and totals lines are not displayed.

-L --license

Display the *gzip* license and quit.

-n --no-name

When compressing, do not save the original file name and time stamp by default. (The original name is always saved if the name had to be truncated.) When decompressing, do not restore the original file name if present (remove only the *gzip* suffix from the compressed file name) and do not restore the original time stamp if present (copy it from the compressed file). This option is the default when decompressing.

-N --name

When compressing, always save the original file name and time stamp; this is the default. When decompressing, restore the original file name and time stamp if present. This option is useful on systems which have a limit on file name length or when the time stamp has been lost after a file transfer.

-q --quiet

Suppress all warnings.

-r --recursive

Travel the directory structure recursively. If any of the file names specified on the command line are directories, *gzip* will descend into the directory and compress all the files it finds there (or decompress them in the case of *gunzip*).

-S .suf --suffix .suf

Use suffix *.suf* instead of *.gz*. Any suffix can be given, but suffixes other than *.z* and *.gz* should be avoided to avoid confusion when files are transferred to other systems. A null suffix forces *gunzip* to try decompression on all given files regardless of suffix, as in:

```
gunzip -S "" *      (*. * for MSDOS)
```

Previous versions of *gzip* used the *.z* suffix. This was changed to avoid a conflict with *pack(1)*.

-t --test

Test. Check the compressed file integrity.

-v --verbose

Verbose. Display the name and percentage reduction for each file compressed or decompressed.

-V --version

Version. Display the version number and compilation options then quit.

-# --fast --best

Regulate the speed of compression using the specified digit #, where **-1** or **--fast** indicates the fastest compression method (less compression) and **-9** or **--best** indicates the slowest compression method (best compression). The default compression level is **-6** (that is, biased towards high compression at expense of speed).

ADVANCED USAGE

Multiple compressed files can be concatenated. In this case, *gunzip* will extract all members at once. For example:

```
gzip -c file1 > foo.gz
gzip -c file2 >> foo.gz
```

Then

```
gunzip -c foo
```

is equivalent to

```
cat file1 file2
```

In case of damage to one member of a .gz file, other members can still be recovered (if the damaged member is removed). However, you can get better compression by compressing all members at once:

```
cat file1 file2 | gzip > foo.gz
```

compresses better than

```
gzip -c file1 file2 > foo.gz
```

If you want to recompress concatenated files to get better compression, do:

```
gzip -cd old.gz | gzip > new.gz
```

If a compressed file consists of several members, the uncompressed size and CRC reported by the --list option applies to the last member only. If you need the uncompressed size for all members, you can use:

```
gzip -cd file.gz | wc -c
```

If you wish to create a single archive file with multiple members so that members can later be extracted independently, use an archiver such as tar or zip. GNU tar supports the -z option to invoke gzip transparently. gzip is designed as a complement to tar, not as a replacement.

ENVIRONMENT

The environment variable **GZIP** can hold a set of default options for *gzip*. These options are interpreted first and can be overwritten by explicit command line parameters. For example:

for sh: `GZIP="-8v --name"; export GZIP`

for csh: `setenv GZIP "-8v --name"`

for MSDOS: `set GZIP=-8v --name`

On Vax/VMS, the name of the environment variable is GZIP_OPT, to avoid a conflict with the symbol set for invocation of the program.

SEE ALSO

znew(1), zcmp(1), zmore(1), zforce(1), gzexe(1), zip(1), unzip(1), compress(1), pack(1), compact(1)

The *gzip* file format is specified in P. Deutsch, GZIP file format specification version 4.3, <<ftp://ftp.isi.edu/in-notes/rfc1952.txt>>, Internet RFC 1952 (May 1996). The *zip* deflation format is specified in P. Deutsch, DEFLATE Compressed Data Format Specification version 1.3, <<ftp://ftp.isi.edu/in-notes/rfc1951.txt>>, Internet RFC 1951 (May 1996).

DIAGNOSTICS

Exit status is normally 0; if an error occurs, exit status is 1. If a warning occurs, exit status is 2.

Usage: `gzip [-cdfhLnNrtvV19] [-S suffix] [file ...]`

Invalid options were specified on the command line.

file: not in gzip format

The file specified to *gunzip* has not been compressed.

file: Corrupt input. Use zcat to recover some data.

The compressed file has been damaged. The data up to the point of failure can be recovered using

```
zcat file > recover
```

file: compressed with *xx* bits, can only handle *yy* bits

File was compressed (using LZW) by a program that could deal with more *bits* than the decompress code on this machine. Recompress the file with gzip, which compresses better and uses less

memory.

file: already has .gz suffix -- no change

The file is assumed to be already compressed. Rename the file and try again.

file already exists; do you wish to overwrite (y or n)?

Respond "y" if you want the output file to be replaced; "n" if not.

gunzip: corrupt input

A SIGSEGV violation was detected which usually means that the input file has been corrupted.

xx.x% Percentage of the input saved by compression.

(Relevant only for **-v** and **-l**.)

-- not a regular file or directory: ignored

When the input file is not a regular file or directory, (e.g. a symbolic link, socket, FIFO, device file), it is left unaltered.

-- has *xx* other links: unchanged

The input file has links; it is left unchanged. See *ln(1)* for more information. Use the **-f** flag to force compression of multiply-linked files.

CAVEATS

When writing compressed data to a tape, it is generally necessary to pad the output with zeroes up to a block boundary. When the data is read and the whole block is passed to *gunzip* for decompression, *gunzip* detects that there is extra trailing garbage after the compressed data and emits a warning by default. You have to use the **--quiet** option to suppress the warning. This option can be set in the **GZIP** environment variable as in:

```
for sh: GZIP="-q" tar -xfz --block-compress /dev/rst0
```

```
for csh: (setenv GZIP -q; tar -xfz --block-compr /dev/rst0
```

In the above example, *gzip* is invoked implicitly by the **-z** option of GNU *tar*. Make sure that the same block size (**-b** option of *tar*) is used for reading and writing compressed data on tapes. (This example assumes you are using the GNU version of *tar*.)

BUGS

The *gzip* format represents the input size modulo 2^{32} , so the **--list** option reports incorrect uncompressed sizes and compression ratios for uncompressed files 4 GB and larger. To work around this problem, you can use the following command to discover a large uncompressed file's true size:

```
zcat file.gz | wc -c
```

The **--list** option reports sizes as **-1** and **crc** as **ffffff** if the compressed file is on a non seekable media.

In some rare cases, the **--best** option gives worse compression than the default compression level (**-6**). On some highly redundant files, *compress* compresses better than *gzip*.

COPYRIGHT NOTICE

Copyright © 1998, 1999, 2001, 2002 Free Software Foundation, Inc.

Copyright © 1992, 1993 Jean-loup Gailly

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

NAME

zcmp, **zdiff** – compare compressed files

SYNOPSIS

zcmp [cmp_options] file1 [file2]

zdiff [diff_options] file1 [file2]

DESCRIPTION

Zcmp and *zdiff* are used to invoke the *cmp* or the *diff* program on files compressed via *gzip*. All options specified are passed directly to *cmp* or *diff*. If only 1 file is specified, then the files compared are *file1* and an uncompressed *file1.gz*. If two files are specified, then they are uncompressed if necessary and fed to *cmp* or *diff*. The exit status from *cmp* or *diff* is preserved.

SEE ALSO

cmp(1), *diff*(1), *zmore*(1), *zgrep*(1), *znew*(1), *zforce*(1), *gzip*(1), *gzexe*(1)

BUGS

Messages from the *cmp* or *diff* programs refer to temporary filenames instead of those specified.

NAME

zgrep – search possibly compressed files for a regular expression

SYNOPSIS

zgrep [*grep_options*] [**-e**] *pattern filename...*

DESCRIPTION

Zgrep invokes *grep* on compressed or gzipped files. All options specified are passed directly to *grep*. If no file is specified, then the standard input is decompressed if necessary and fed to *grep*. Otherwise the given files are uncompressed if necessary and fed to *grep*.

If the GREP environment variable is set, *zgrep* uses it as the *grep* program to be invoked.

AUTHOR

Charles Levert (charles@comm.polymtl.ca)

SEE ALSO

grep(1), *gzexe*(1), *gzip*(1), *zdiff*(1), *zforce*(1), *zmore*(1), *znew*(1)

NAME

zip – package and compress (archive) files

SYNOPSIS

zip [-aABcdDeEffghjklLmoqrRSTuvVwXyz!@\$] [--longoption ...] [-b path] [-n suffixes] [-t date] [-tt date] [*zipfile* [*file* ...]] [-xi list]

zipcloak (see separate man page)

zipnote (see separate man page)

zipsplit (see separate man page)

Note: Command line processing in *zip* has been changed to support long options and handle all options and arguments more consistently. Some old command lines that depend on command line inconsistencies may no longer work.

DESCRIPTION

zip is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows 9x/NT/XP, Minix, Atari, Macintosh, Amiga, and Acorn RISC OS. It is analogous to a combination of the Unix commands *tar*(1) and *compress*(1) and is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems).

A companion program (*unzip*(1L)) unpacks *zip* archives. The *zip* and *unzip*(1L) programs can work with archives produced by PKZIP (supporting most PKZIP features up to PKZIP version 4.6), and PKZIP and PKUNZIP can work with archives produced by *zip* (with some exceptions, notably streamed archives, but recent changes in the *zip* file standard may facilitate better compatibility). *zip* version 3.0 is compatible with PKZIP 2.04 and also supports the Zip64 extensions of PKZIP 4.5 which allow archives as well as files to exceed the previous 2 GB limit (4 GB in some cases). *zip* also now supports **bzip2** compression if the **bzip2** library is included when *zip* is compiled. Note that PKUNZIP 1.10 cannot extract files produced by PKZIP 2.04 or *zip* 3.0. You must use PKUNZIP 2.04g or *unzip* 5.0p1 (or later versions) to extract them.

See the **EXAMPLES** section at the bottom of this page for examples of some typical uses of *zip*.

Large Archives and Zip64. *zip* automatically uses the Zip64 extensions when files larger than 4 GB are added to an archive, an archive containing Zip64 entries is updated (if the resulting archive still needs Zip64), the size of the archive will exceed 4 GB, or when the number of entries in the archive will exceed about 64K. Zip64 is also used for archives streamed from standard input as the size of such archives are not known in advance, but the option **-fz-** can be used to force *zip* to create PKZIP 2 compatible archives (as long as Zip64 extensions are not needed). You must use a PKZIP 4.5 compatible *unzip*, such as *unzip* 6.0 or later, to extract files using the Zip64 extensions.

In addition, streamed archives, entries encrypted with standard encryption, or split archives created with the pause option may not be compatible with PKZIP as data descriptors are used and PKZIP at the time of this writing does not support data descriptors (but recent changes in the PKWare published *zip* standard now include some support for the data descriptor format *zip* uses).

Mac OS X. Though previous Mac versions had their own *zip* port, *zip* supports Mac OS X as part of the Unix port and most Unix features apply. References to "MacOS" below generally refer to MacOS versions older than OS X. Support for some Mac OS features in the Unix Mac OS X port, such as resource forks, is expected in the next *zip* release.

For a brief help on *zip* and *unzip*, run each without specifying any parameters on the command line.

USE

The program is useful for packaging a set of files for distribution; for archiving files; and for saving disk space by temporarily compressing unused files or directories.

The *zip* program puts one or more compressed files into a single *zip* archive, along with information about the files (name, path, date, time of last modification, protection, and check information to verify file integrity). An entire directory structure can be packed into a *zip* archive with a single command.

Compression ratios of 2:1 to 3:1 are common for text files. *zip* has one compression method (deflation) and can also store files without compression. (If **bzip2** support is added, *zip* can also compress using **bzip2** compression, but such entries require a reasonably modern unzip to decompress. When **bzip2** compression is selected, it replaces deflation as the default method.) *zip* automatically chooses the better of the two (deflation or store or, if **bzip2** is selected, **bzip2** or store) for each file to be compressed.

Command format. The basic command format is

```
zip options archive inpath inpath ...
```

where **archive** is a new or existing *zip* archive and **inpath** is a directory or file path optionally including wildcards. When given the name of an existing *zip* archive, *zip* will replace identically named entries in the *zip* archive (matching the relative names as stored in the archive) or add entries for new names. For example, if *foo.zip* exists and contains *foo/file1* and *foo/file2*, and the directory *foo* contains the files *foo/file1* and *foo/file3*, then:

```
zip -r foo.zip foo
```

or more concisely

```
zip -r foo foo
```

will replace *foo/file1* in *foo.zip* and add *foo/file3* to *foo.zip*. After this, *foo.zip* contains *foo/file1*, *foo/file2*, and *foo/file3*, with *foo/file2* unchanged from before.

So if before the *zip* command is executed *foo.zip* has:

```
foo/file1 foo/file2
```

and directory *foo* has:

```
file1 file3
```

then *foo.zip* will have:

```
foo/file1 foo/file2 foo/file3
```

where *foo/file1* is replaced and *foo/file3* is new.

-@ file lists. If a file list is specified as **-@** [Not on MacOS], *zip* takes the list of input files from standard input instead of from the command line. For example,

```
zip -@ foo
```

will store the files listed one per line on stdin in *foo.zip*.

Under Unix, this option can be used to powerful effect in conjunction with the *find* (1) command. For example, to archive all the C source files in the current directory and its subdirectories:

```
find . -name "*.c" -print | zip source -@
```

(note that the pattern must be quoted to keep the shell from expanding it).

Streaming input and output. *zip* will also accept a single dash ("-") as the *zip* file name, in which case it will write the *zip* file to standard output, allowing the output to be piped to another program. For example:

```
zip -r - . | dd of=/dev/nrst0 obs=16k
```

would write the *zip* output directly to a tape with the specified block size for the purpose of backing up the current directory.

zip also accepts a single dash ("-") as the name of a file to be compressed, in which case it will read the file from standard input, allowing *zip* to take input from another program. For example:

```
tar cf - . | zip backup -
```

would compress the output of the *tar* command for the purpose of backing up the current directory. This generally produces better compression than the previous example using the **-r** option because *zip* can take advantage of redundancy between files. The backup can be restored using the command

```
unzip -p backup | tar xf -
```

When no *zip* file name is given and stdout is not a terminal, *zip* acts as a filter, compressing standard input to standard output. For example,

```
tar cf - . | zip | dd of=/dev/nrst0 obs=16k
```

is equivalent to

```
tar cf - . | zip - - | dd of=/dev/nrst0 obs=16k
```

zip archives created in this manner can be extracted with the program *funzip* which is provided in the *unzip* package, or by *gunzip* which is provided in the *gzip* package (but some *gunzip* may not support this if *zip* used the Zip64 extensions). For example:

```
dd if=/dev/nrst0 ibs=16k | funzip | tar xvf -
```

The stream can also be saved to a file and *unzip* used.

If Zip64 support for large files and archives is enabled and *zip* is used as a filter, *zip* creates a Zip64 archive that requires a PKZIP 4.5 or later compatible unzip to read it. This is to avoid ambiguities in the zip file structure as defined in the current zip standard (PKWARE AppNote) where the decision to use Zip64 needs to be made before data is written for the entry, but for a stream the size of the data is not known at that point. If the data is known to be smaller than 4 GB, the option **-fz-** can be used to prevent use of Zip64, but *zip* will exit with an error if Zip64 was in fact needed. *zip* 3 and *unzip* 6 and later can read archives with Zip64 entries. Also, *zip* removes the Zip64 extensions if not needed when archive entries are copied (see the **-U** (**--copy**) option).

When directing the output to another file, note that all options should be before the redirection including **-x**. For example:

```
zip archive "*.h" "*.c" -x donotinclude.h orthis.h > tofile
```

Zip files. When changing an existing *zip* archive, *zip* will write a temporary file with the new contents, and only replace the old one when the process of creating the new version has been completed without error.

If the name of the *zip* archive does not contain an extension, the extension **.zip** is added. If the name already contains an extension other than **.zip**, the existing extension is kept unchanged. However, split archives (archives split over multiple files) require the **.zip** extension on the last split.

Scanning and reading files. When *zip* starts, it scans for files to process (if needed). If this scan takes longer than about 5 seconds, *zip* will display a "Scanning files" message and start displaying progress dots every 2 seconds or every so many entries processed, whichever takes longer. If there is more than 2 seconds between dots it could indicate that finding each file is taking time and could mean a slow network connection for example. (Actually the initial file scan is a two-step process where the directory scan is followed by a sort and these two steps are separated with a space in the dots. If updating an existing archive, a space also appears between the existing file scan and the new file scan.) The scanning files dots are not controlled by the **-ds** dot size option, but the dots are turned off by the **-q** quiet option. The **-sf** show files option can be used to scan for files and get the list of files scanned without actually processing them.

If *zip* is not able to read a file, it issues a warning but continues. See the **-MM** option below for more on how *zip* handles patterns that are not matched and files that are not readable. If some files were skipped, a warning is issued at the end of the zip operation noting how many files were read and how many skipped.

Command modes. *zip* now supports two distinct types of command modes, **external** and **internal**. The **external** modes (add, update, and freshen) read files from the file system (as well as from an existing archive) while the **internal** modes (delete and copy) operate exclusively on entries in an existing archive.

add

Update existing entries and add new files. If the archive does not exist create it. This is the default mode.

update (-u)

Update existing entries if newer on the file system and add new files. If the archive does not exist issue warning then create a new archive.

freshen (-f)

Update existing entries of an archive if newer on the file system. Does not add new files to the archive.

delete (-d)

Select entries in an existing archive and delete them.

copy (-U)

Select entries in an existing archive and copy them to a new archive. This new mode is similar to **update** but command line patterns select entries in the existing archive rather than files from the file system and it uses the **--out** option to write the resulting archive to a new file rather than update the existing archive, leaving the original archive unchanged.

The new File Sync option (**-FS**) is also considered a new mode, though it is similar to **update**. This mode synchronizes the archive with the files on the OS, only replacing files in the archive if the file time or size of the OS file is different, adding new files, and deleting entries from the archive where there is no matching file. As this mode can delete entries from the archive, consider making a backup copy of the archive.

Also see **-DF** for creating difference archives.

See each option description below for details and the **EXAMPLES** section below for examples.

Split archives. *zip* version 3.0 and later can create split archives. A **split archive** is a standard zip archive split over multiple files. (Note that split archives are not just archives split in to pieces, as the offsets of entries are now based on the start of each split. Concatenating the pieces together will invalidate these offsets, but *unzip* can usually deal with it. *zip* will usually refuse to process such a spliced archive unless the **-FF** fix option is used to fix the offsets.)

One use of split archives is storing a large archive on multiple removable media. For a split archive with 20 split files the files are typically named (replace ARCHIVE with the name of your archive) ARCHIVE.z01, ARCHIVE.z02, ..., ARCHIVE.z19, ARCHIVE.zip. Note that the last file is the **.zip** file. In contrast, **spanned archives** are the original multi-disk archive generally requiring floppy disks and using volume labels to store disk numbers. *zip* supports split archives but not spanned archives, though a procedure exists for converting split archives of the right size to spanned archives. The reverse is also true, where each file of a spanned archive can be copied in order to files with the above names to create a split archive.

Use **-s** to set the split size and create a split archive. The size is given as a number followed optionally by one of k (kB), m (MB), g (GB), or t (TB) (the default is m). The **-sp** option can be used to pause *zip* between splits to allow changing removable media, for example, but read the descriptions and warnings for both **-s** and **-sp** below.

Though *zip* does not update split archives, *zip* provides the new option **-O** (**--output-file** or **--out**) to allow split archives to be updated and saved in a new archive. For example,

```
zip inarchive.zip foo.c bar.c --out outarchive.zip
```

reads archive **inarchive.zip**, even if split, adds the files **foo.c** and **bar.c**, and writes the resulting archive to **outarchive.zip**. If **inarchive.zip** is split then **outarchive.zip** defaults to the same split size. Be aware that if **outarchive.zip** and any split files that are created with it already exist, these are always overwritten as needed without warning. This may be changed in the future.

Unicode. Though the zip standard requires storing paths in an archive using a specific character set, in practice zips have stored paths in archives in whatever the local character set is. This creates problems when an archive is created or updated on a system using one character set and then extracted on another system using a different character set. When compiled with Unicode support enabled on platforms that support wide characters, *zip* now stores, in addition to the standard local path for backward compatibility, the UTF-8 translation of the path. This provides a common universal character set for storing paths that allows these paths to be fully extracted on other systems that support Unicode and to match as close as possible on systems that don't.

On Win32 systems where paths are internally stored as Unicode but represented in the local character set, it's possible that some paths will be skipped during a local character set directory scan. *zip* with Unicode support now can read and store these paths. Note that Win 9x systems and FAT file systems don't fully support Unicode.

Be aware that console windows on Win32 and Unix, for example, sometimes don't accurately show all characters due to how each operating system switches in character sets for display. However, directory navigation tools should show the correct paths if the needed fonts are loaded.

Command line format. This version of *zip* has updated command line processing and support for long options.

Short options take the form

```
-s[-][s[-]...] [value] [=value] [ value]
```

where *s* is a one or two character short option. A short option that takes a value is last in an argument and anything after it is taken as the value. If the option can be negated and "-" immediately follows the option, the option is negated. Short options can also be given as separate arguments

```
-s[-] [value] [=value] [ value] -s[-] [value] [=value] [ value] ...
```

Short options in general take values either as part of the same argument or as the following argument. An optional = is also supported. So

```
-ttmmddyyyy
```

and

```
-tt=mmddyyyy
```

and

```
-tt mmddyyyy
```

all work. The **-x** and **-i** options accept lists of values and use a slightly different format described below. See the **-x** and **-i** options.

Long options take the form

```
--longoption[-] [=value] [ value]
```

where the option starts with --, has a multicharacter name, can include a trailing dash to negate the option (if the option supports it), and can have a value (option argument) specified by preceeding it with = (no spaces). Values can also follow the argument. So

```
--before-date=mmddyyyy
```

and

```
--before-date mmddyyyy
```

both work.

Long option names can be shortened to the shortest unique abbreviation. See the option descriptions below for which support long options. To avoid confusion, avoid abbreviating a negatable option with an embedded dash ("-") at the dash if you plan to negate it (the parser would consider a trailing dash, such as for the option **--some-option** using **--some-** as the option, as part of the name rather than a negating dash). This may be changed to force the last dash in **--some-** to be negating in the future.

OPTIONS

-a

--ascii [Systems using EBCDIC] Translate file to ASCII format.

-A

--adjust-sfx

Adjust self-extracting executable archive. A self-extracting executable archive is created by prepending the SFX stub to an existing archive. The **-A** option tells *zip* to adjust the entry offsets stored in the archive to take into account this "preamble" data.

Note: self-extracting archives for the Amiga are a special case. At present, only the Amiga port of *zip* is capable of adjusting or updating these without corrupting them. **-J** can be used to remove the SFX stub if other updates need to be made.

-AC**--archive-clear**

[WIN32] Once archive is created (and tested if **-T** is used, which is recommended), clear the archive bits of files processed. WARNING: Once the bits are cleared they are cleared. You may want to use the **-sf** show files option to store the list of files processed in case the archive operation must be repeated. Also consider using the **-MM** must match option. Be sure to check out **-DF** as a possibly better way to do incremental backups.

-AS**--archive-set**

[WIN32] Only include files that have the archive bit set. Directories are not stored when **-AS** is used, though by default the paths of entries, including directories, are stored as usual and can be used by most unzips to recreate directories.

The archive bit is set by the operating system when a file is modified and, if used with **-AC**, **-AS** can provide an incremental backup capability. However, other applications can modify the archive bit and it may not be a reliable indicator of which files have changed since the last archive operation. Alternative ways to create incremental backups are using **-t** to use file dates, though this won't catch old files copied to directories being archived, and **-DF** to create a differential archive.

-B**--binary**

[VM/CMS and MVS] force file to be read binary (default is text).

-Bn

[TANDEM] set Edit/Enscribe formatting options with *n* defined as
 bit 0: Don't add delimiter (Edit/Enscribe)
 bit 1: Use LF rather than CR/LF as delimiter (Edit/Enscribe)
 bit 2: Space fill record to maximum record length (Enscribe)
 bit 3: Trim trailing space (Enscribe)
 bit 8: Force 30K (Expand) large read for unstructured files

-b path**--temp-path path**

Use the specified *path* for the temporary *zip* archive. For example:

```
zip -b /tmp stuff *
```

will put the temporary *zip* archive in the directory */tmp*, copying over *stuff.zip* to the current directory when done. This option is useful when updating an existing archive and the file system containing this old archive does not have enough space to hold both old and new archives at the same time. It may also be useful when streaming in some cases to avoid the need for data descriptors. Note that using this option may require *zip* take additional time to copy the archive file when done to the destination file system.

-c**--entry-comments**

Add one-line comments for each file. File operations (adding, updating) are done first, and the user is then prompted for a one-line comment for each file. Enter the comment followed by return, or just return for no comment.

-C**--preserve-case**

[VMS] Preserve case all on VMS. Negating this option (**-C-**) downcases.

-C2**--preserve-case-2**

[VMS] Preserve case ODS2 on VMS. Negating this option (**-C2-**) downcases.

-C5**--preserve-case-5**

[VMS] Preserve case ODS5 on VMS. Negating this option (**-C5-**) downcases.

-d**--delete**

Remove (delete) entries from a *zip* archive. For example:

```
zip -d foo foo/tom/junk foo/harry/\* \*.o
```

will remove the entry *foo/tom/junk*, all of the files that start with *foo/harry/*, and all of the files that end with *.o* (in any path). Note that shell pathname expansion has been inhibited with backslashes, so that *zip* can see the asterisks, enabling *zip* to match on the contents of the *zip* archive instead of the contents of the current directory. (The backslashes are not used on MSDOS-based platforms.) Can also use quotes to escape the asterisks as in

```
zip -d foo foo/tom/junk "foo/harry/*" "*.o"
```

Not escaping the asterisks on a system where the shell expands wildcards could result in the asterisks being converted to a list of files in the current directory and that list used to delete entries from the archive.

Under MSDOS, **-d** is case sensitive when it matches names in the *zip* archive. This requires that file names be entered in upper case if they were zipped by PKZIP on an MSDOS system. (We considered making this case insensitive on systems where paths were case insensitive, but it is possible the archive came from a system where case does matter and the archive could include both **Bar** and **bar** as separate files in the archive.) But see the new option **-ic** to ignore case in the archive.

-db**--display-bytes**

Display running byte counts showing the bytes zipped and the bytes to go.

-dc**--display-counts**

Display running count of entries zipped and entries to go.

-dd**--display-dots**

Display dots while each entry is zipped (except on ports that have their own progress indicator). See **-ds** below for setting dot size. The default is a dot every 10 MB of input file processed. The

-v option also displays dots (previously at a much higher rate than this but now **-v** also defaults to 10 MB) and this rate is also controlled by **-ds**.

-df

--datafork

[MacOS] Include only data-fork of files zipped into the archive. Good for exporting files to foreign operating-systems. Resource-forks will be ignored at all.

-dg

--display-globaldots

Display progress dots for the archive instead of for each file. The command

```
zip -qdgds 10m
```

will turn off most output except dots every 10 MB.

-ds size

--dot-size size

Set amount of input file processed for each dot displayed. See **-dd** to enable displaying dots. Setting this option implies **-dd**. Size is in the format nm where n is a number and m is a multiplier. Currently m can be k (KB), m (MB), g (GB), or t (TB), so if n is 100 and m is k, size would be 100k which is 100 KB. The default is 10 MB.

The **-v** option also displays dots and now defaults to 10 MB also. This rate is also controlled by this option. A size of 0 turns dots off.

This option does not control the dots from the "Scanning files" message as *zip* scans for input files. The dot size for that is fixed at 2 seconds or a fixed number of entries, whichever is longer.

-du

--display-usize

Display the uncompressed size of each entry.

-dv

--display-volume

Display the volume (disk) number each entry is being read from, if reading an existing archive, and being written to.

-D

--no-dir-entries

Do not create entries in the *zip* archive for directories. Directory entries are created by default so that their attributes can be saved in the zip archive. The environment variable ZIPOPT can be used to change the default options. For example under Unix with sh:

```
ZILOPT="-D"; export ZIPOPT
```

(The variable ZIPOPT can be used for any option, including **-i** and **-x** using a new option format detailed below, and can include several options.) The option **-D** is a shorthand for **-x "*/"** but the latter previously could not be set as default in the ZIPOPT environment variable as the contents of ZIPOPT gets inserted near the beginning of the command line and the file list had to end at the end of the line.

This version of *zip* does allow **-x** and **-i** options in ZIPOPT if the form

```
-x file file ... @
```

is used, where the @ (an argument that is just @) terminates the list.

-DF

--difference-archive

Create an archive that contains all new and changed files since the original archive was created. For this to work, the input file list and current directory must be the same as during the original *zip* operation.

For example, if the existing archive was created using

```
zip -r foofull .
```

from the *bar* directory, then the command

```
zip -r foofull . -DF --out foonew
```

also from the *bar* directory creates the archive *foonew* with just the files not in *foofull* and the files where the size or file time of the files do not match those in *foofull*.

Note that the timezone environment variable TZ should be set according to the local timezone in order for this option to work correctly. A change in timezone since the original archive was created could result in no times matching and all files being included.

A possible approach to backing up a directory might be to create a normal archive of the contents of the directory as a full backup, then use this option to create incremental backups.

-e

--encrypt

Encrypt the contents of the *zip* archive using a password which is entered on the terminal in response to a prompt (this will not be echoed; if standard error is not a tty, *zip* will exit with an error). The password prompt is repeated to save the user from typing errors.

-E

--longnames

[OS/2] Use the .LONGNAME Extended Attribute (if found) as filename.

-f

--freshen

Replace (freshen) an existing entry in the *zip* archive only if it has been modified more recently than the version already in the *zip* archive; unlike the update option (**-u**) this will not add files that are not already in the *zip* archive. For example:

```
zip -f foo
```

This command should be run from the same directory from which the original *zip* command was run, since paths stored in *zip* archives are always relative.

Note that the timezone environment variable TZ should be set according to the local timezone in order for the **-f**, **-u** and **-o** options to work correctly.

The reasons behind this are somewhat subtle but have to do with the differences between the Unix-format file times (always in GMT) and most of the other operating systems (always local time) and the necessity to compare the two. A typical TZ value is "MET-IMEST" (Middle European time with automatic adjustment for "summertime" or Daylight Savings Time).

The format is TTThhDDD, where TTT is the time zone such as MET, hh is the difference between GMT and local time such as -1 above, and DDD is the time zone when daylight savings time is in effect. Leave off the DDD if there is no daylight savings time. For the US Eastern time zone EST5EDT.

-F
--fix
-FF
--fixfix

Fix the *zip* archive. The **-F** option can be used if some portions of the archive are missing, but requires a reasonably intact central directory. The input archive is scanned as usual, but *zip* will ignore some problems. The resulting archive should be valid, but any inconsistent entries will be left out.

When doubled as in **-FF**, the archive is scanned from the beginning and *zip* scans for special signatures to identify the limits between the archive members. The single **-F** is more reliable if the archive is not too much damaged, so try this option first.

If the archive is too damaged or the end has been truncated, you must use **-FF**. This is a change from *zip* 2.32, where the **-F** option is able to read a truncated archive. The **-F** option now more reliably fixes archives with minor damage and the **-FF** option is needed to fix archives where **-F** might have been sufficient before.

Neither option will recover archives that have been incorrectly transferred in ascii mode instead of binary. After the repair, the **-t** option of *unzip* may show that some files have a bad CRC. Such files cannot be recovered; you can remove them from the archive using the **-d** option of *zip*.

Note that **-FF** may have trouble fixing archives that include an embedded zip archive that was stored (without compression) in the archive and, depending on the damage, it may find the entries in the embedded archive rather than the archive itself. Try **-F** first as it does not have this problem.

The format of the fix commands have changed. For example, to fix the damaged archive *foo.zip*,

```
zip -F foo --out foofix
```

tries to read the entries normally, copying good entries to the new archive *foofix.zip*. If this doesn't work, as when the archive is truncated, or if some entries you know are in the archive are missed, then try

```
zip -FF foo --out foofixfix
```

and compare the resulting archive to the archive created by **-F**. The **-FF** option may create an inconsistent archive. Depending on what is damaged, you can then use the **-F** option to fix that archive.

A split archive with missing split files can be fixed using **-F** if you have the last split of the archive (the **.zip** file). If this file is missing, you must use **-FF** to fix the archive, which will prompt you for the splits you have.

Currently the fix options can't recover entries that have a bad checksum or are otherwise damaged.

-FI

--fifo [Unix] Normally *zip* skips reading any FIFOs (named pipes) encountered, as *zip* can hang if the FIFO is not being fed. This option tells *zip* to read the contents of any FIFO it finds.

-FS

--filesync

Synchronize the contents of an archive with the files on the OS. Normally when an archive is updated, new files are added and changed files are updated but files that no longer exist on the OS are not deleted from the archive. This option enables a new mode that checks entries in the archive against the file system. If the file time and file size of the entry matches that of the OS file, the entry is copied from the old archive instead of being read from the file system and compressed. If the OS file has changed, the entry is read and compressed as usual. If the entry in the archive does not match a file on the OS, the entry is deleted. Enabling this option should create archives

that are the same as new archives, but since existing entries are copied instead of compressed, updating an existing archive with **-FS** can be much faster than creating a new archive. Also consider using **-u** for updating an archive.

For this option to work, the archive should be updated from the same directory it was created in so the relative paths match. If few files are being copied from the old archive, it may be faster to create a new archive instead.

Note that the timezone environment variable **TZ** should be set according to the local timezone in order for this option to work correctly. A change in timezone since the original archive was created could result in no times matching and recompression of all files.

This option deletes files from the archive. If you need to preserve the original archive, make a copy of the archive first or use the **--out** option to output the updated archive to a new file. Even though it may be slower, creating a new archive with a new archive name is safer, avoids mismatches between archive and OS paths, and is preferred.

-g

--grow

Grow (append to) the specified *zip* archive, instead of creating a new one. If this operation fails, *zip* attempts to restore the archive to its original state. If the restoration fails, the archive might become corrupted. This option is ignored when there's no existing archive or when at least one archive member must be updated or deleted.

-h

-?

--help

Display the *zip* help information (this also appears if *zip* is run with no arguments).

-h2

--more-help

Display extended help including more on command line format, pattern matching, and more obscure options.

-i files

--include files

Include only the specified files, as in:

```
zip -r foo . -i \*.c
```

which will include only the files that end in *.c* in the current directory and its subdirectories. (Note for PKZIP users: the equivalent command is

```
pkzip -rP foo *.c
```

PKZIP does not allow recursion in directories other than the current one.) The backslash avoids the shell filename substitution, so that the name matching is performed by *zip* at all directory levels. [This is for Unix and other systems where ** escapes the next character. For other systems where the shell does not process *** do not use ** and the above is

```
zip -r foo . -i *.c
```

Examples are for Unix unless otherwise specified.] So to include *dir*, a directory directly under the current directory, use

```
zip -r foo . -i dir/\*
```

or

```
zip -r foo . -i "dir/*"
```

to match paths such as `dir/a` and `dir/b/file.c` [on ports without wildcard expansion in the shell such as MSDOS and Windows]

```
zip -r foo . -i dir/*
```

is used.] Note that currently the trailing `/` is needed for directories (as in

```
zip -r foo . -i dir/
```

to include directory `dir`).

The long option form of the first example is

```
zip -r foo . --include \*.c
```

and does the same thing as the short option form.

Though the command syntax used to require **-i** at the end of the command line, this version actually allows **-i** (or **--include**) anywhere. The list of files terminates at the next argument starting with `-`, the end of the command line, or the list terminator `@` (an argument that is just `@`). So the above can be given as

```
zip -i \*.c @ -r foo .
```

for example. There must be a space between the option and the first file of a list. For just one file you can use the single value form

```
zip -i \*.c -r foo .
```

(no space between option and value) or

```
zip --include=\*.c -r foo .
```

as additional examples. The single value forms are not recommended because they can be confusing and, in particular, the **-ifile** format can cause problems if the first letter of **file** combines with **i** to form a two-letter option starting with **i**. Use **-sc** to see how your command line will be parsed.

Also possible:

```
zip -r foo . -i@include.lst
```

which will only include the files in the current directory and its subdirectories that match the patterns in the file `include.lst`.

Files to **-i** and **-x** are patterns matching internal archive paths. See **-R** for more on patterns.

-I

--no-image

[Acorn RISC OS] Don't scan through Image files. When used, *zip* will not consider Image files (eg. DOS partitions or Spark archives when SparkFS is loaded) as directories but will store them as single files.

For example, if you have SparkFS loaded, zipping a Spark archive will result in a zipfile containing a directory (and its content) while using the **'I'** option will result in a zipfile containing a Spark archive. Obviously this second case will also be obtained (without the **'I'** option) if SparkFS isn't loaded.

-ic

--ignore-case

[VMS, WIN32] Ignore case when matching archive entries. This option is only available on systems where the case of files is ignored. On systems with case-insensitive file systems, case is normally ignored when matching files on the file system but is not ignored for **-f** (freshen), **-d** (delete), **-U** (copy), and similar modes when matching against archive entries (currently **-f** ignores case on

VMS) because archive entries can be from systems where case does matter and names that are the same except for case can exist in an archive. The **-ic** option makes all matching case insensitive. This can result in multiple archive entries matching a command line pattern.

-j

--junk-paths

Store just the name of a saved file (junk the path), and do not store directory names. By default, *zip* will store the full path (relative to the current directory).

-jj

--absolute-path

[MacOS] record Fullpath (+ Volname). The complete path including volume will be stored. By default the relative path will be stored.

-J

--junk-sfx

Strip any prepended data (e.g. a SFX stub) from the archive.

-k

--DOS-names

Attempt to convert the names and paths to conform to MSDOS, store only the MSDOS attribute (just the user write attribute from Unix), and mark the entry as made under MSDOS (even though it was not); for compatibility with PKUNZIP under MSDOS which cannot handle certain names such as those with two dots.

-l

--to-crlf

Translate the Unix end-of-line character LF into the MSDOS convention CR LF. This option should not be used on binary files. This option can be used on Unix if the zip file is intended for PKUNZIP under MSDOS. If the input files already contain CR LF, this option adds an extra CR. This is to ensure that **unzip -a** on Unix will get back an exact copy of the original file, to undo the effect of **zip -l**. See **-ll** for how binary files are handled.

-la

--log-append

Append to existing logfile. Default is to overwrite.

-lf logfilepath

--logfile-path logfilepath

Open a logfile at the given path. By default any existing file at that location is overwritten, but the **-la** option will result in an existing file being opened and the new log information appended to any existing information. Only warnings and errors are written to the log unless the **-li** option is also given, then all information messages are also written to the log.

-li

--log-info

Include information messages, such as file names being zipped, in the log. The default is to only include the command line, any warnings and errors, and the final status.

-ll

--from-crlf

Translate the MSDOS end-of-line CR LF into Unix LF. This option should not be used on binary files. This option can be used on MSDOS if the zip file is intended for unzip under Unix. If the file is converted and the file is later determined to be binary a warning is issued and the file is probably corrupted. In this release if **-ll** detects binary in the first buffer read from a file, *zip* now issues a warning and skips line end conversion on the file. This check seems to catch all binary files tested, but the original check remains and if a converted file is later determined to be binary

that warning is still issued. A new algorithm is now being used for binary detection that should allow line end conversion of text files in **UTF-8** and similar encodings.

-L

--license

Display the *zip* license.

-m

--move

Move the specified files into the *zip* archive; actually, this deletes the target directories/files after making the specified *zip* archive. If a directory becomes empty after removal of the files, the directory is also removed. No deletions are done until *zip* has created the archive without error. This is useful for conserving disk space, but is potentially dangerous so it is recommended to use it in combination with **-T** to test the archive before removing all input files.

-MM

--must-match

All input patterns must match at least one file and all input files found must be readable. Normally when an input pattern does not match a file the "name not matched" warning is issued and when an input file has been found but later is missing or not readable a missing or not readable warning is issued. In either case *zip* continues creating the archive, with missing or unreadable new files being skipped and files already in the archive remaining unchanged. After the archive is created, if any files were not readable *zip* returns the OPEN error code (18 on most systems) instead of the normal success return (0 on most systems). With **-MM** set, *zip* exits as soon as an input pattern is not matched (whenever the "name not matched" warning would be issued) or when an input file is not readable. In either case *zip* exits with an OPEN error and no archive is created.

This option is useful when a known list of files is to be zipped so any missing or unreadable files will result in an error. It is less useful when used with wildcards, but *zip* will still exit with an error if any input pattern doesn't match at least one file and if any matched files are unreadable. If you want to create the archive anyway and only need to know if files were skipped, don't use **-MM** and just check the return code. Also **-If** could be useful.

-n suffixes

--suffixes suffixes

Do not attempt to compress files named with the given **suffixes**. Such files are simply stored (0% compression) in the output zip file, so that *zip* doesn't waste its time trying to compress them. The suffixes are separated by either colons or semicolons. For example:

```
zip -rn .Z:.zip:.tiff:.gif:.snd foo foo
```

will copy everything from *foo* into *foo.zip*, but will store any files that end in *.Z*, *.zip*, *.tiff*, *.gif*, or *.snd* without trying to compress them (image and sound files often have their own specialized compression methods). By default, *zip* does not compress files with extensions in the list *.Z:.zip:.zoo:.arc:.lzh:.arj*. Such files are stored directly in the output archive. The environment variable **ZIPOPT** can be used to change the default options. For example under Unix with *csh*:

```
setenv ZIPOPT "-n .gif:.zip"
```

To attempt compression on all files, use:

```
zip -n : foo
```

The maximum compression option **-9** also attempts compression on all files regardless of extension.

On Acorn RISC OS systems the suffixes are actually filetypes (3 hex digit format). By default, *zip* does not compress files with filetypes in the list *DDC:D96:68E* (i.e. Archives, CFS files and Pack-Dir files).

-nw

--no-wild

Do not perform internal wildcard processing (shell processing of wildcards is still done by the shell unless the arguments are escaped). Useful if a list of paths is being read and no wildcard substitution is desired.

-N**--notes**

[Amiga, MacOS] Save Amiga or MacOS filenotes as zipfile comments. They can be restored by using the -N option of *unzip*. If -c is used also, you are prompted for comments only for those files that do not have filenotes.

-o**--latest-time**

Set the "last modified" time of the *zip* archive to the latest (oldest) "last modified" time found among the entries in the *zip* archive. This can be used without any other operations, if desired. For example:

```
zip -o foo
```

will change the last modified time of **foo.zip** to the latest time of the entries in **foo.zip**.

-O output-file**--output-file output-file**

Process the archive changes as usual, but instead of updating the existing archive, output the new archive to output-file. Useful for updating an archive without changing the existing archive and the input archive must be a different file than the output archive.

This option can be used to create updated split archives. It can also be used with **-U** to copy entries from an existing archive to a new archive. See the **EXAMPLES** section below.

Another use is converting *zip* files from one split size to another. For instance, to convert an archive with 700 MB CD splits to one with 2 GB DVD splits, can use:

```
zip -s 2g cd-split.zip --out dvd-split.zip
```

which uses copy mode. See **-U** below. Also:

```
zip -s 0 split.zip --out unsplit.zip
```

will convert a split archive to a single-file archive.

Copy mode will convert stream entries (using data descriptors and which should be compatible with most unzips) to normal entries (which should be compatible with all unzips), except if standard encryption was used. For archives with encrypted entries, *zipcloak* will decrypt the entries and convert them to normal entries.

-p**--paths**

Include relative file paths as part of the names of files stored in the archive. This is the default. The **-j** option junks the paths and just stores the names of the files.

-P password**--password password**

Use *password* to encrypt zipfile entries (if any). **THIS IS INSECURE!** Many multi-user operating systems provide ways for any user to see the current command line of any other user; even on stand-alone systems there is always the threat of over-the-shoulder peeking. Storing the plaintext password as part of a command line in an automated script is even worse. Whenever possible, use the non-echoing, interactive prompt to enter passwords. (And where security is truly important, use strong encryption such as Pretty Good Privacy instead of the relatively weak standard encryption provided by zipfile utilities.)

-q**--quiet**

Quiet mode; eliminate informational messages and comment prompts. (Useful, for example, in shell scripts and background tasks).

-Qn**--Q-flag n**

[QDOS] store information about the file in the file header with *n* defined as

bit 0: Don't add headers for any file

bit 1: Add headers for all files

bit 2: Don't wait for interactive key press on exit

-r**--recurse-paths**

Travel the directory structure recursively; for example:

```
zip -r foo.zip foo
```

or more concisely

```
zip -r foo foo
```

In this case, all the files and directories in **foo** are saved in a *zip* archive named **foo.zip**, including files with names starting with ".", since the recursion does not use the shell's file-name substitution mechanism. If you wish to include only a specific subset of the files in directory **foo** and its subdirectories, use the **-i** option to specify the pattern of files to be included. You should not use **-r** with the name ".*", since that matches "." which will attempt to zip up the parent directory (probably not what was intended).

Multiple source directories are allowed as in

```
zip -r foo foo1 foo2
```

which first zips up **foo1** and then **foo2**, going down each directory.

Note that while wildcards to **-r** are typically resolved while recursing down directories in the file system, any **-R**, **-x**, and **-i** wildcards are applied to internal archive pathnames once the directories are scanned. To have wildcards apply to files in subdirectories when recursing on Unix and similar systems where the shell does wildcard substitution, either escape all wildcards or put all arguments with wildcards in quotes. This lets *zip* see the wildcards and match files in subdirectories using them as it recurses.

-R**--recurse-patterns**

Travel the directory structure recursively starting at the current directory; for example:

```
zip -R foo "*.c"
```

In this case, all the files matching ***.c** in the tree starting at the current directory are stored into a *zip* archive named **foo.zip**. Note that ***.c** will match **file.c**, **a/file.c** and **a/b/c**. More than one pattern can be listed as separate arguments. Note for PKZIP users: the equivalent command is

```
pkzip -rP foo *.c
```

Patterns are relative file paths as they appear in the archive, or will after zipping, and can have optional wildcards in them. For example, given the current directory is **foo** and under it are directories **foo1** and **foo2** and in **foo1** is the file **bar.c**,

```
zip -R foo/*
```

will zip up **foo**, **foo/foo1**, **foo/foo1/bar.c**, and **foo/foo2**.

```
zip -R */bar.c
```

will zip up **foo/foo1/bar.c**. See the note for **-r** on escaping wildcards.

-RE**--regex**

[WIN32] Before *zip* 3.0, regular expression list matching was enabled by default on Windows platforms. Because of confusion resulting from the need to escape "[" and "]" in names, it is now off by default for Windows so "[" and "]" are just normal characters in names. This option enables [] matching again.

-s splitsize**--split-size** splitsize

Enable creating a split archive and set the split size. A split archive is an archive that could be split over many files. As the archive is created, if the size of the archive reaches the specified split size, that split is closed and the next split opened. In general all splits but the last will be the split size and the last will be whatever is left. If the entire archive is smaller than the split size a single-file archive is created.

Split archives are stored in numbered files. For example, if the output archive is named **archive** and three splits are required, the resulting archive will be in the three files **archive.z01**, **archive.z02**, and **archive.zip**. Do not change the numbering of these files or the archive will not be readable as these are used to determine the order the splits are read.

Split size is a number optionally followed by a multiplier. Currently the number must be an integer. The multiplier can currently be one of **k** (kilobytes), **m** (megabytes), **g** (gigabytes), or **t** (terabytes). As 64k is the minimum split size, numbers without multipliers default to megabytes. For example, to create a split archive called **foo** with the contents of the **bar** directory with splits of 670 MB that might be useful for burning on CDs, the command:

```
zip -s 670m -r foo bar
```

could be used.

Currently the old splits of a split archive are not excluded from a new archive, but they can be specifically excluded. If possible, keep the input and output archives out of the path being zipped when creating split archives.

Using **-s** without **-sp** as above creates all the splits where **foo** is being written, in this case the current directory. This split mode updates the splits as the archive is being created, requiring all splits to remain writable, but creates split archives that are readable by any unzip that supports split archives. See **-sp** below for enabling split pause mode which allows splits to be written directly to removable media.

The option **-sv** can be used to enable verbose splitting and provide details of how the splitting is being done. The **-sb** option can be used to ring the bell when *zip* pauses for the next split destination.

Split archives cannot be updated, but see the **-O** (**--out**) option for how a split archive can be updated as it is copied to a new archive. A split archive can also be converted into a single-file archive using a split size of 0 or negating the **-s** option:

```
zip -s 0 split.zip --out single.zip
```

Also see **-U** (**--copy**) for more on using copy mode.

-sb**--split-bell**

If splitting and using split pause mode, ring the bell when *zip* pauses for each split destination.

-sc

--show-command

Show the command line starting *zip* as processed and exit. The new command parser permutes the arguments, putting all options and any values associated with them before any non-option arguments. This allows an option to appear anywhere in the command line as long as any values that go with the option go with it. This option displays the command line as *zip* sees it, including any arguments from the environment such as from the **ZIPOPT** variable. Where allowed, options later in the command line can override options earlier in the command line.

-sf**--show-files**

Show the files that would be operated on, then exit. For instance, if creating a new archive, this will list the files that would be added. If the option is negated, **-sf-**, output only to an open log file. Screen display is not recommended for large lists.

-so**--show-options**

Show all available options supported by *zip* as compiled on the current system. As this command reads the option table, it should include all options. Each line includes the short option (if defined), the long option (if defined), the format of any value that goes with the option, if the option can be negated, and a small description. The value format can be no value, required value, optional value, single character value, number value, or a list of values. The output of this option is not intended to show how to use any option but only show what options are available.

-sp**--split-pause**

If splitting is enabled with **-s**, enable split pause mode. This creates split archives as **-s** does, but stream writing is used so each split can be closed as soon as it is written and *zip* will pause between each split to allow changing split destination or media.

Though this split mode allows writing splits directly to removable media, it uses stream archive format that may not be readable by some unzips. Before relying on splits created with **-sp**, test a split archive with the unzip you will be using.

To convert a stream split archive (created with **-sp**) to a standard archive see the **--out** option.

-su**--show-unicode**

As **-sf**, but also show Unicode version of the path if exists.

-sU**--show-just-unicode**

As **-sf**, but only show Unicode version of the path if exists, otherwise show the standard version of the path.

-sv**--split-verbose**

Enable various verbose messages while splitting, showing how the splitting is being done.

-S**--system-hidden**

[MSDOS, OS/2, WIN32 and ATARI] Include system and hidden files.

[MacOS] Includes finder invisible files, which are ignored otherwise.

-t mmddyyyy**--from-date mmddyyyy**

Do not operate on files modified prior to the specified date, where **mm** is the month (00-12), **dd** is the day of the month (01-31), and **yyyy** is the year. The *ISO 8601* date format **yyyy-mm-dd** is also accepted. For example:

```
zip -rt 12071991 infamy foo
```

```
zip -rt 1991-12-07 infamy foo
```

will add all the files in **foo** and its subdirectories that were last modified on or after 7 December 1991, to the *zip* archive **infamy.zip**.

-tt mmddyyyy

--before-date mmddyyyy

Do not operate on files modified after or at the specified date, where **mm** is the month (00-12), **dd** is the day of the month (01-31), and **yyyy** is the year. The *ISO 8601* date format **yyyy-mm-dd** is also accepted. For example:

```
zip -rtt 11301995 infamy foo
```

```
zip -rtt 1995-11-30 infamy foo
```

will add all the files in **foo** and its subdirectories that were last modified before 30 November 1995, to the *zip* archive **infamy.zip**.

-T

--test

Test the integrity of the new zip file. If the check fails, the old zip file is unchanged and (with the **-m** option) no input files are removed.

-TT cmd

--unzip-command cmd

Use command *cmd* instead of 'unzip -tqq' to test an archive when the **-T** option is used. On Unix, to use a copy of unzip in the current directory instead of the standard system unzip, could use:

```
zip archive file1 file2 -T -TT "./unzip -tqq"
```

In *cmd*, { } is replaced by the name of the temporary archive, otherwise the name of the archive is appended to the end of the command. The return code is checked for success (0 on Unix).

-u

--update

Replace (update) an existing entry in the *zip* archive only if it has been modified more recently than the version already in the *zip* archive. For example:

```
zip -u stuff *
```

will add any new files in the current directory, and update any files which have been modified since the *zip* archive *stuff.zip* was last created/modified (note that *zip* will not try to pack *stuff.zip* into itself when you do this).

Note that the **-u** option with no input file arguments acts like the **-f** (freshen) option.

-U

--copy-entries

Copy entries from one archive to another. Requires the **--out** option to specify a different output file than the input archive. Copy mode is the reverse of **-d** delete. When delete is being used with **--out**, the selected entries are deleted from the archive and all other entries are copied to the new archive, while copy mode selects the files to include in the new archive. Unlike **-u** update, input patterns on the command line are matched against archive entries only and not the file system files. For instance,

```
zip inarchive "*.c" --copy --out outarchive
```

copies entries with names ending in **.c** from **inarchive** to **outarchive**. The wildcard must be escaped on some systems to prevent the shell from substituting names of files from the file system which may have no relevance to the entries in the archive.

If no input files appear on the command line and **--out** is used, copy mode is assumed:

```
zip inarchive --out outarchive
```

This is useful for changing split size for instance. Encrypting and decrypting entries is not yet supported using copy mode. Use *zipcloak* for that.

-UN v

--unicode v

Determine what *zip* should do with Unicode file names. *zip 3.0*, in addition to the standard file path, now includes the UTF-8 translation of the path if the entry path is not entirely 7-bit ASCII. When an entry is missing the Unicode path, *zip* reverts back to the standard file path. The problem with using the standard path is this path is in the local character set of the *zip* that created the entry, which may contain characters that are not valid in the character set being used by the *unzip*. When *zip* is reading an archive, if an entry also has a Unicode path, *zip* now defaults to using the Unicode path to recreate the standard path using the current local character set.

This option can be used to determine what *zip* should do with this path if there is a mismatch between the stored standard path and the stored UTF-8 path (which can happen if the standard path was updated). In all cases, if there is a mismatch it is assumed that the standard path is more current and *zip* uses that. Values for **v** are

- q – quit if paths do not match
- w – warn, continue with standard path
- i – ignore, continue with standard path
- n – no Unicode, do not use Unicode paths

The default is to warn and continue.

Characters that are not valid in the current character set are escaped as **#Uxxxx** and **#Lxxxxxx**, where **x** is an ASCII character for a hex digit. The first is used if a 16-bit character number is sufficient to represent the Unicode character and the second if the character needs more than 16 bits to represent its Unicode character code. Setting **-UN** to

e – escape

as in

```
zip archive -sU -UN=e
```

forces *zip* to escape all characters that are not printable 7-bit ASCII.

Normally *zip* stores UTF-8 directly in the standard path field on systems where UTF-8 is the current character set and stores the UTF-8 in the new extra fields otherwise. The option

u – UTF-8

as in

```
zip archive dir -r -UN=UTF8
```

forces *zip* to store UTF-8 as native in the archive. Note that storing UTF-8 directly is the default on Unix systems that support it. This option could be useful on Windows systems where the escaped path is too large to be a valid path and the UTF-8 version of the path is smaller, but native UTF-8 is not backward compatible on Windows systems.

-v

--verbose

Verbose mode or print diagnostic version info.

Normally, when applied to real operations, this option enables the display of a progress indicator during compression (see **-dd** for more on dots) and requests verbose diagnostic info about zipfile structure oddities.

However, when **-v** is the only command line argument a diagnostic screen is printed instead. This should now work even if stdout is redirected to a file, allowing easy saving of the information for sending with bug reports to Info-ZIP. The version screen provides the help screen header with program name, version, and release date, some pointers to the Info-ZIP home and distribution sites, and shows information about the target environment (compiler type and version, OS version, compilation date and the enabled optional features used to create the *zip* executable).

-V

--VMS-portable

[VMS] Save VMS file attributes. (Files are truncated at EOF.) When a **-V** archive is unpacked on a non-VMS system, some file types (notably Stream_LF text files and pure binary files like fixed-512) should be extracted intact. Indexed files and file types with embedded record sizes (notably variable-length record types) will probably be seen as corrupt elsewhere.

-VV

--VMS-specific

[VMS] Save VMS file attributes, and all allocated blocks in a file, including any data beyond EOF. Useful for moving ill-formed files among VMS systems. When a **-VV** archive is unpacked on a non-VMS system, almost all files will appear corrupt.

-w

--VMS-versions

[VMS] Append the version number of the files to the name, including multiple versions of files. Default is to use only the most recent version of a specified file.

-ww

--VMS-dot-versions

[VMS] Append the version number of the files to the name, including multiple versions of files, using the .nnn format. Default is to use only the most recent version of a specified file.

-ws

--wild-stop-dirs

Wildcards match only at a directory level. Normally *zip* handles paths as strings and given the paths

`/foo/bar/dir/file1.c`

`/foo/bar/file2.c`

an input pattern such as

`/foo/bar/*`

normally would match both paths, the ***** matching **dir/file1.c** and **file2.c**. Note that in the first case a directory boundary (*/*) was crossed in the match. With **-ws** no directory bounds will be included in the match, making wildcards local to a specific directory level. So, with **-ws** enabled, only the second path would be matched.

When using **-ws**, use ****** to match across directory boundaries as ***** does normally.

-x files

--exclude files

Explicitly exclude the specified files, as in:

`zip -r foo foo -x *.o`

which will include the contents of **foo** in **foo.zip** while excluding all the files that end in **.o**. The backslash avoids the shell filename substitution, so that the name matching is performed by *zip* at all directory levels.

Also possible:

```
zip -r foo foo -x@exclude.lst
```

which will include the contents of **foo** in **foo.zip** while excluding all the files that match the patterns in the file **exclude.lst**.

The long option forms of the above are

```
zip -r foo foo --exclude \*.o
```

and

```
zip -r foo foo --exclude @exclude.lst
```

Multiple patterns can be specified, as in:

```
zip -r foo foo -x \*.o \*.c
```

If there is no space between **-x** and the pattern, just one value is assumed (no list):

```
zip -r foo foo -x\*.o
```

See **-i** for more on include and exclude.

-X

--no-extra

Do not save extra file attributes (Extended Attributes on OS/2, uid/gid and file times on Unix). The zip format uses extra fields to include additional information for each entry. Some extra fields are specific to particular systems while others are applicable to all systems. Normally when *zip* reads entries from an existing archive, it reads the extra fields it knows, strips the rest, and adds the extra fields applicable to that system. With **-X**, *zip* strips all old fields and only includes the Unicode and Zip64 extra fields (currently these two extra fields cannot be disabled).

Negating this option, **-X-**, includes all the default extra fields, but also copies over any unrecognized extra fields.

-y

--symlinks

For UNIX and VMS (V8.3 and later), store symbolic links as such in the *zip* archive, instead of compressing and storing the file referred to by the link. This can avoid multiple copies of files being included in the archive as *zip* recurses the directory trees and accesses files directly and by links.

-z

--archive-comment

Prompt for a multi-line comment for the entire *zip* archive. The comment is ended by a line containing just a period, or an end of file condition (^D on Unix, ^Z on MSDOS, OS/2, and VMS). The comment can be taken from a file:

```
zip -z foo < foowhat
```

-Z cm

--compression-method cm

Set the default compression method. Currently the main methods supported by *zip* are **store** and **deflate**. Compression method can be set to:

store – Setting the compression method to **store** forces *zip* to store entries with no compression. This is generally faster than compressing entries, but results in no space savings. This is the same as using **-0** (compression level zero).

deflate – This is the default method for *zip*. If *zip* determines that storing is better than deflation, the entry will be stored instead.

bzip2 – If **bzip2** support is compiled in, this compression method also becomes available. Only some modern unzips currently support the **bzip2** compression method, so test the unzip you will be using before relying on archives using this method (compression method 12).

For example, to add **bar.c** to archive **foo** using **bzip2** compression:

```
zip -Z bzip2 foo bar.c
```

The compression method can be abbreviated:

```
zip -Zb foo bar.c
```

-#

(-0, -1, -2, -3, -4, -5, -6, -7, -8, -9)

Regulate the speed of compression using the specified digit #, where **-0** indicates no compression (store all files), **-1** indicates the fastest compression speed (less compression) and **-9** indicates the slowest compression speed (optimal compression, ignores the suffix list). The default compression level is **-6**.

Though still being worked, the intention is this setting will control compression speed for all compression methods. Currently only deflation is controlled.

#!

--use-privileges

[WIN32] Use privileges (if granted) to obtain all aspects of WinNT security.

--@

--names-stdin

Take the list of input files from standard input. Only one filename per line.

-\$

--volume-label

[MSDOS, OS/2, WIN32] Include the volume label for the drive holding the first file to be compressed. If you want to include only the volume label or to force a specific drive, use the drive name as first file name, as in:

```
zip -$ foo a: c:bar
```

EXAMPLES

The simplest example:

```
zip stuff *
```

creates the archive *stuff.zip* (assuming it does not exist) and puts all the files in the current directory in it, in compressed form (the **.zip** suffix is added automatically, unless the archive name contains a dot already; this allows the explicit specification of other suffixes).

Because of the way the shell on Unix does filename substitution, files starting with "." are not included; to include these as well:

```
zip stuff .* *
```

Even this will not include any subdirectories from the current directory.

To zip up an entire directory, the command:

```
zip -r foo foo
```

creates the archive *foo.zip*, containing all the files and directories in the directory *foo* that is contained within the current directory.

You may want to make a *zip* archive that contains the files in *foo*, without recording the directory name, *foo*. You can use the **-j** option to leave off the paths, as in:

```
zip -j foo foo/*
```


If you are short on disk space, you might not have enough room to hold both the original directory and the corresponding compressed *zip* archive. In this case, you can create the archive in steps using the **-m** option. If *foo* contains the subdirectories *tom*, *dick*, and *harry*, you can:

```
zip -rm foo foo/tom
zip -rm foo foo/dick
zip -rm foo foo/harry
```

where the first command creates *foo.zip*, and the next two add to it. At the completion of each *zip* command, the last created archive is deleted, making room for the next *zip* command to function.

Use **-s** to set the split size and create a split archive. The size is given as a number followed optionally by one of k (kB), m (MB), g (GB), or t (TB). The command

```
zip -s 2g -r split.zip foo
```

creates a split archive of the directory *foo* with splits no bigger than 2 GB each. If *foo* contained 5 GB of contents and the contents were stored in the split archive without compression (to make this example simple), this would create three splits, *split.z01* at 2 GB, *split.z02* at 2 GB, and *split.zip* at a little over 1 GB.

The **-sp** option can be used to pause *zip* between splits to allow changing removable media, for example, but read the descriptions and warnings for both **-s** and **-sp** below.

Though *zip* does not update split archives, *zip* provides the new option **-O** (**--output-file**) to allow split archives to be updated and saved in a new archive. For example,

```
zip inarchive.zip foo.c bar.c --out outarchive.zip
```

reads archive **inarchive.zip**, even if split, adds the files **foo.c** and **bar.c**, and writes the resulting archive to **outarchive.zip**. If **inarchive.zip** is split then **outarchive.zip** defaults to the same split size. Be aware that **outarchive.zip** and any split files that are created with it are always overwritten without warning. This may be changed in the future.

PATTERN MATCHING

This section applies only to Unix. Watch this space for details on MSDOS and VMS operation. However, the special wildcard characters ***** and **[]** below apply to at least MSDOS also.

The Unix shells (*sh*, *csh*, *bash*, and others) normally do filename substitution (also called "globbing") on command arguments. Generally the special characters are:

- ?** match any single character
- *** match any number of characters (including none)
- []** match any character in the range indicated within the brackets (example: [a-f], [0-9]). This form of wildcard matching allows a user to specify a list of characters between square brackets and if any of the characters match the expression matches. For example:

```
zip archive "*. [hc] "
```

would archive all files in the current directory that end in **.h** or **.c**.

Ranges of characters are supported:

```
zip archive "[a-f] *"
```

would add to the archive all files starting with "a" through "f".

Negation is also supported, where any character in that position not in the list matches. Negation is supported by adding `!` or `^` to the beginning of the list:

```
zip archive "*. [!o]"
```

matches files that don't end in ".o".

On WIN32, `[]` matching needs to be turned on with the `-RE` option to avoid the confusion that names with `[` or `]` have caused.

When these characters are encountered (without being escaped with a backslash or quotes), the shell will look for files relative to the current path that match the pattern, and replace the argument with a list of the names that matched.

The *zip* program can do the same matching on names that are in the *zip* archive being modified or, in the case of the `-x` (exclude) or `-i` (include) options, on the list of files to be operated on, by using backslashes or quotes to tell the shell not to do the name expansion. In general, when *zip* encounters a name in the list of files to do, it first looks for the name in the file system. If it finds it, it then adds it to the list of files to do. If it does not find it, it looks for the name in the *zip* archive being modified (if it exists), using the pattern matching characters described above, if present. For each match, it will add that name to the list of files to be processed, unless this name matches one given with the `-x` option, or does not match any name given with the `-i` option.

The pattern matching includes the path, and so patterns like `*.o` match names that end in ".o", no matter what the path prefix is. Note that the backslash must precede every special character (i.e. `?*[]`), or the entire argument must be enclosed in double quotes (`"`).

In general, use backslashes or double quotes for paths that have wildcards to make *zip* do the pattern matching for file paths, and always for paths and strings that have spaces or wildcards for `-i`, `-x`, `-R`, `-d`, and `-U` and anywhere *zip* needs to process the wildcards.

ENVIRONMENT

The following environment variables are read and used by *zip* as described.

ZIPOPT

contains default options that will be used when running *zip*. The contents of this environment variable will get added to the command line just after the **zip** command.

ZIP [Not on RISC OS and VMS] see ZIPOPT

Zip\$Options

[RISC OS] see ZIPOPT

Zip\$Exts

[RISC OS] contains extensions separated by a `:` that will cause native filenames with one of the specified extensions to be added to the zip file with basename and extension swapped.

ZIP_OPTS

[VMS] see ZIPOPT

SEE ALSO

compress(1), shar(1L), tar(1), unzip(1L), gzip(1L)

DIAGNOSTICS

The exit status (or error level) approximates the exit codes defined by PKWARE and takes on the following values, except under VMS:

- 0 normal; no errors or warnings detected.
- 2 unexpected end of zip file.
- 3 a generic error in the zipfile format was detected. Processing may have completed successfully anyway; some broken zipfiles created by other archivers have simple work-arounds.

- 4 *zip* was unable to allocate memory for one or more buffers during program initialization.
- 5 a severe error in the zipfile format was detected. Processing probably failed immediately.
- 6 entry too large to be processed (such as input files larger than 2 GB when not using Zip64 or trying to read an existing archive that is too large) or entry too large to be split with *zipsplit*
- 7 invalid comment format
- 8 *zip -T* failed or out of memory
- 9 the user aborted *zip* prematurely with control-C (or similar)
- 10 *zip* encountered an error while using a temp file
- 11 read or seek error
- 12 *zip* has nothing to do
- 13 missing or empty zip file
- 14 error writing to a file
- 15 *zip* was unable to create a file to write to
- 16 bad command line parameters
- 18 *zip* could not open a specified file to read
- 19 *zip* was compiled with options not supported on this system

VMS interprets standard Unix (or PC) return values as other, scarier-looking things, so *zip* instead maps them into VMS-style status codes. In general, *zip* sets VMS Facility = 1955 (0x07A3), Code = 2* Unix_status, and an appropriate Severity (as specified in ziperr.h). More details are included in the VMS-specific documentation. See [.vms]NOTES.TXT and [.vms]vms_msg_gen.c.

BUGS

zip 3.0 is not compatible with PKUNZIP 1.10. Use *zip* 1.1 to produce *zip* files which can be extracted by PKUNZIP 1.10.

zip files produced by *zip* 3.0 must not be *updated* by *zip* 1.1 or PKZIP 1.10, if they contain encrypted members or if they have been produced in a pipe or on a non-seekable device. The old versions of *zip* or PKZIP would create an archive with an incorrect format. The old versions can list the contents of the zip file but cannot extract it anyway (because of the new compression algorithm). If you do not use encryption and use regular disk files, you do not have to care about this problem.

Under VMS, not all of the odd file formats are treated properly. Only stream-LF format *zip* files are expected to work with *zip*. Others can be converted using Rahul Dhesi's BILF program. This version of *zip* handles some of the conversion internally. When using Kermit to transfer zip files from VMS to MSDOS, type "set file type block" on VMS. When transferring from MSDOS to VMS, type "set file type fixed" on VMS. In both cases, type "set file type binary" on MSDOS.

Under some older VMS versions, *zip* may hang for file specifications that use DECnet syntax *foo::*.**.

On OS/2, *zip* cannot match some names, such as those including an exclamation mark or a hash sign. This is a bug in OS/2 itself: the 32-bit DosFindFirst/Next don't find such names. Other programs such as GNU tar are also affected by this bug.

Under OS/2, the amount of Extended Attributes displayed by DIR is (for compatibility) the amount returned by the 16-bit version of DosQueryPathInfo(). Otherwise OS/2 1.3 and 2.0 would report different EA sizes when DIRing a file. However, the structure layout returned by the 32-bit DosQueryPathInfo() is a bit different, it uses extra padding bytes and link pointers (it's a linked list) to have all fields on 4-byte boundaries for portability to future RISC OS/2 versions. Therefore the value reported by *zip* (which uses this 32-bit-mode size) differs from that reported by DIR. *zip* stores the 32-bit format for portability, even the 16-bit MS-C-compiled version running on OS/2 1.3, so even this one shows the 32-bit-mode size.

AUTHORS

Copyright (C) 1997-2008 Info-ZIP.

Currently distributed under the Info-ZIP license.

Copyright (C) 1990-1997 Mark Adler, Richard B. Wales, Jean-loup Gailly, Onno van der Linden, Kai Uwe Rommel, Igor Mandrichenko, John Bush and Paul Kienitz.

Original copyright:

Permission is granted to any individual or institution to use, copy, or redistribute this software so long as all of the original files are included, that it is not sold for profit, and that this copyright notice is retained.

LIKE ANYTHING ELSE THAT'S FREE, ZIP AND ITS ASSOCIATED UTILITIES ARE PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

Please send bug reports and comments using the web page at: www.info-zip.org. For bug reports, please include the version of *zip* (see *zip -h*), the make options used to compile it (see *zip -v*), the machine and operating system in use, and as much additional information as possible.

ACKNOWLEDGEMENTS

Thanks to R. P. Byrne for his *Shrink.Pas* program, which inspired this project, and from which the shrink algorithm was stolen; to Phil Katz for placing in the public domain the *zip* file format, compression format, and .ZIP filename extension, and for accepting minor changes to the file format; to Steve Burg for clarifications on the deflate format; to Haruhiko Okumura and Leonid Broukhis for providing some useful ideas for the compression algorithm; to Keith Petersen, Rich Wales, Hunter Goatley and Mark Adler for providing a mailing list and *ftp* site for the Info-ZIP group to use; and most importantly, to the Info-ZIP group itself (listed in the file *infozip.who*) without whose tireless testing and bug-fixing efforts a portable *zip* would not have been possible. Finally we should thank (blame) the first Info-ZIP moderator, David Kirschbaum, for getting us into this mess in the first place. The manual page was rewritten for Unix by R. P. C. Rodgers and updated by E. Gordon for *zip* 3.0.

NAME

zless – file perusal filter for crt viewing of compressed text

SYNOPSIS

zless [name ...]

DESCRIPTION

Zless is a filter which allows examination of compressed or plain text files one screenful at a time on a soft-copy terminal. It is the equivalent of setting the environment variable LESSOPEN to '|gzip -cdfq -- %s', and the environment variable LESSMETACHARS to and then running *less*. However, enough people seem to think that having the command *zless* available is important to be worth providing it.

SEE ALSO

zmore(1), *less*(1)

BUGS

Zless does not work with compressed data that is piped to it via standard input; it requires that input files be specified as arguments.

COPYRIGHT NOTICE

Copyright © 2006 Free Software Foundation, Inc.

Copyright © 1992, 1993 Jean-loup Gailly

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

NAME

zmore – file perusal filter for crt viewing of compressed text

SYNOPSIS

zmore [name ...]

DESCRIPTION

Zmore is a filter which allows examination of compressed or plain text files one screenful at a time on a soft-copy terminal. *zmore* works on files compressed with *compress*, *pack* or *gzip*, and also on uncompressed files. If a file does not exist, *zmore* looks for a file of the same name with the addition of a .gz, .z or .Z suffix.

Zmore normally pauses after each screenful, printing --More-- at the bottom of the screen. If the user then types a carriage return, one more line is displayed. If the user hits a space, another screenful is displayed. Other possibilities are enumerated later.

Zmore looks in the file */etc/termcap* to determine terminal characteristics, and to determine the default window size. On a terminal capable of displaying 24 lines, the default window size is 22 lines. To use a pager other than the default *more*, set environment variable *PAGER* to the name of the desired program, such as *less*.

Other sequences which may be typed when *zmore* pauses, and their effects, are as follows (*i* is an optional integer argument, defaulting to 1) :

- i*<space> display *i* more lines, (or another screenful if no argument is given)
- ^D display 11 more lines (a “scroll”). If *i* is given, then the scroll size is set to *i*.
- d same as ^D (control-D)
- iz* same as typing a space except that *i*, if present, becomes the new window size. Note that the window size reverts back to the default at the end of the current file.
- is* skip *i* lines and print a screenful of lines
- if* skip *i* screenfuls and print a screenful of lines
- q or Q quit reading the current file; go on to the next (if any)
- e or q When the prompt --More--(Next file: *file*) is printed, this command causes *zmore* to exit.
- s When the prompt --More--(Next file: *file*) is printed, this command causes *zmore* to skip the next file and continue.
- = Display the current line number.
- i*/expr search for the *i*-th occurrence of the regular expression *expr*. If the pattern is not found, *zmore* goes on to the next file (if any). Otherwise, a screenful is displayed, starting two lines before the place where the expression was found. The user's erase and kill characters may be used to edit the regular expression. Erasing back past the first column cancels the search command.
- in* search for the *i*-th occurrence of the last regular expression entered.
- !command invoke a shell with *command*. The character '!' in "command" is replaced with the previous shell command. The sequence "\!" is replaced by "!".
- :q or :Q quit reading the current file; go on to the next (if any) (same as q or Q).
- . (dot) repeat the previous command.

The commands take effect immediately, i.e., it is not necessary to type a carriage return. Up to the time when the command character itself is given, the user may hit the line kill character to cancel the numerical argument being formed. In addition, the user may hit the erase character to redisplay the --More-- message.

At any time when output is being sent to the terminal, the user can hit the quit key (normally control-\\).

Zmore will stop sending output, and will display the usual --More-- prompt. The user may then enter one of the above commands in the normal manner. Unfortunately, some output is lost when this is done, due to the fact that any characters waiting in the terminal's output queue are flushed when the quit signal occurs.

The terminal is set to *noecho* mode by this program so that the output can be continuous. What you type will thus not show on your terminal, except for the / and ! commands.

If the standard output is not a teletype, then *zmore* acts just like *zcat*, except that a header is printed before each file.

FILES

/etc/termcap
Terminal data base

SEE ALSO

more(1), gzip(1), zdiff(1), zgrep(1), znew(1), zforce(1), gzexe(1)

NAME

zstd – zstd, zstdmt, unzstd, zstdcat – Compress or decompress .zst files

SYNOPSIS

zstd [*OPTIONS*] [-*INPUT-FILE*] [-o *OUTPUT-FILE*]

zstdmt is equivalent to **zstd -T0**

unzstd is equivalent to **zstd -d**

zstdcat is equivalent to **zstd -dcf**

DESCRIPTION

zstd is a fast lossless compression algorithm and data compression tool, with command line syntax similar to **gzip** (1) and **xz** (1). It is based on the **LZ77** family, with further FSE & huff0 entropy stages. **zstd** offers highly configurable compression speed, with fast modes at > 200 MB/s per core, and strong modes nearing lzma compression ratios. It also features a very fast decoder, with speeds > 500 MB/s per core.

zstd command line syntax is generally similar to gzip, but features the following differences :

- Source files are preserved by default. It's possible to remove them automatically by using the **--rm** command.
- When compressing a single file, **zstd** displays progress notifications and result summary by default. Use **-q** to turn them off.
- **zstd** does not accept input from console, but it properly accepts **stdin** when it's not the console.
- **zstd** displays a short help page when command line is an error. Use **-q** to turn it off.

zstd compresses or decompresses each *file* according to the selected operation mode. If no *files* are given or *file* is -, **zstd** reads from standard input and writes the processed data to standard output. **zstd** will refuse to write compressed data to standard output if it is a terminal : it will display an error message and skip the *file*. Similarly, **zstd** will refuse to read compressed data from standard input if it is a terminal.

Unless **--stdout** or **-o** is specified, *files* are written to a new file whose name is derived from the source *file* name:

- When compressing, the suffix **.zst** is appended to the source filename to get the target filename.
- When decompressing, the **.zst** suffix is removed from the source filename to get the target filename

Concatenation with .zst files

It is possible to concatenate **.zst** files as is. **zstd** will decompress such files as if they were a single **.zst** file.

OPTIONS**Integer suffixes and special values**

In most places where an integer argument is expected, an optional suffix is supported to easily indicate large integers. There must be no space between the integer and the suffix.

KiB Multiply the integer by 1,024 (2¹⁰). **Ki**, **K**, and **KB** are accepted as synonyms for **KiB**.

MiB Multiply the integer by 1,048,576 (2²⁰). **Mi**, **M**, and **MB** are accepted as synonyms for **MiB**.

Operation mode

If multiple operation mode options are given, the last one takes effect.

-z, --compress

Compress. This is the default operation mode when no operation mode option is specified and no other operation mode is implied from the command name (for example, **unzstd** implies **--decompress**).

-d, --decompress, --uncompress

Decompress.

-t, --test

Test the integrity of compressed *files*. This option is equivalent to **--decompress --stdout** except that the decompressed data is discarded instead of being written to standard output. No files are created or removed.

-b# Benchmark file(s) using compression level #

--train FILES

Use FILES as a training set to create a dictionary. The training set should contain a lot of small files (> 100).

-l, --list

Display information related to a zstd compressed file, such as size, ratio, and checksum. Some of these fields may not be available. This command can be augmented with the **-v** modifier.

Operation modifiers

- **-#**: # compression level [1–19] (default: 3)
- **--ultra**: unlocks high compression levels 20+ (maximum 22), using a lot more memory. Note that decompression will also require more memory when using these levels.
- **--fast[=#]**: switch to ultra-fast compression levels. If **=#** is not present, it defaults to **1**. The higher the value, the faster the compression speed, at the cost of some compression ratio. This setting overwrites compression level if one was set previously. Similarly, if a compression level is set after **--fast**, it overrides it.
- **-T#**, **--threads=#**: Compress using # working threads (default: 1). If # is 0, attempt to detect and use the number of physical CPU cores. In all cases, the nb of threads is capped to **ZSTDMT_NBWORKERS_MAX**, which is either 64 in 32-bit mode, or 256 for 64-bit environments. This modifier does nothing if **zstd** is compiled without multithread support.
- **--single-thread**: Does not spawn a thread for compression, use a single thread for both I/O and compression. In this mode, compression is serialized with I/O, which is slightly slower. (This is different from **-T1**, which spawns 1 compression thread in parallel of I/O). This mode is the only one available when multithread support is disabled. Single-thread mode features lower memory usage. Final compressed result is slightly different from **-T1**.
- **--adapt[=min=#,max=#]** : **zstd** will dynamically adapt compression level to perceived I/O conditions. Compression level adaptation can be observed live by using command **-v**. Adaptation can be constrained between supplied **min** and **max** levels. The feature works when combined with multi-threading and **--long** mode. It does not work with **--single-thread**. It sets window size to 8 MB by default (can be changed manually, see **wlog**). Due to the chaotic nature of dynamic adaptation, compressed result is not reproducible. *note* : at the time of this writing, **--adapt** can remain stuck at low speed when combined with multiple worker threads (>=2).
- **--long[=#]**: enables long distance matching with # **windowLog**, if not # is not present it defaults to **27**. This increases the window size (**windowLog**) and memory usage for both the compressor and decompressor. This setting is designed to improve the compression ratio for files with long matches at a large distance.

Note: If **windowLog** is set to larger than 27, **--long=windowLog** or **--memory=windowSize** needs to be passed to the decompressor.

- **-D DICT**: use **DICT** as Dictionary to compress or decompress FILE(s)
- **--patch-from FILE**: Specify the file to be used as a reference point for zstd's diff engine. This is effectively dictionary compression with some convenient parameter selection, namely that window-Size > srcSize.

Note: cannot use both this and `-D` together Note: `--long` mode will be automatically activated if `chainLog < fileLog` (`fileLog` being the `windowLog` required to cover the whole file). You can also manually force it. Note: for all levels, you can use `--patch-from` in `--single-thread` mode to improve compression ratio at the cost of speed Note: for level 19, you can get increased compression ratio at the cost of speed by specifying `--zstd=targetLength=` to be something large (i.e 4096), and by setting a large `--zstd=chainLog=`

- `--rsyncable` : `zstd` will periodically synchronize the compression state to make the compressed file more rsync-friendly. There is a negligible impact to compression ratio, and the faster compression levels will see a small compression speed hit. This feature does not work with `--single-thread`. You probably don't want to use it with long range mode, since it will decrease the effectiveness of the synchronization points, but your mileage may vary.
- `-C`, `--[no-]check`: add integrity check computed from uncompressed data (default: enabled)
- `--[no-]content-size`: enable / disable whether or not the original size of the file is placed in the header of the compressed file. The default option is `--content-size` (meaning that the original size will be placed in the header).
- `--no-dictID`: do not store dictionary ID within frame header (dictionary compression). The decoder will have to rely on implicit knowledge about which dictionary to use, it won't be able to check if it's correct.
- `-M#`, `--memory=#`: Set a memory usage limit. By default, Zstandard uses 128 MB for decompression as the maximum amount of memory the decompressor is allowed to use, but you can override this manually if need be in either direction (ie. you can increase or decrease it).

This is also used during compression when using with `--patch-from=`. In this case, this parameter overrides that maximum size allowed for a dictionary. (128 MB).

- `--stream-size=#` : Sets the pledged source size of input coming from a stream. This value must be exact, as it will be included in the produced frame header. Incorrect stream sizes will cause an error. This information will be used to better optimize compression parameters, resulting in better and potentially faster compression, especially for smaller source sizes.
- `--size-hint=#`: When handling input from a stream, `zstd` must guess how large the source size will be when optimizing compression parameters. If the stream size is relatively small, this guess may be a poor one, resulting in a higher compression ratio than expected. This feature allows for controlling the guess when needed. Exact guesses result in better compression ratios. Overestimates result in slightly degraded compression ratios, while underestimates may result in significant degradation.
- `-o FILE`: save result into **FILE**
- `-f`, `--force`: disable input and output checks. Allows overwriting existing files, input from console, output to stdout, operating on links, block devices, etc.
- `-c`, `--stdout`: force write to standard output, even if it is the console
- `--[no-]sparse`: enable / disable sparse FS support, to make files with many zeroes smaller on disk. Creating sparse files may save disk space and speed up decompression by reducing the amount of disk I/O. default: enabled when output is into a file, and disabled when output is stdout. This setting overrides default and can force sparse mode over stdout.
- `--rm`: remove source file(s) after successful compression or decompression. If used in combination with `-o`, will trigger a confirmation prompt (which can be silenced with `-f`), as this is a destructive operation.
- `-k`, `--keep`: keep source file(s) after successful compression or decompression. This is the default behavior.
- `-r`: operate recursively on directories
- `--filelist FILE` read a list of files to process as content from **FILE**. Format is compatible with `ls` output, with one file per line.

- **--output-dir-flat DIR**: resulting files are stored into target **DIR** directory, instead of same directory as origin file. Be aware that this command can introduce name collision issues, if multiple files, from different directories, end up having the same name. Collision resolution ensures first file with a given name will be present in **DIR**, while in combination with **-f**, the last file will be present instead.
- **--output-dir-mirror DIR**: similar to **--output-dir-flat**, the output files are stored underneath target **DIR** directory, but this option will replicate input directory hierarchy into output **DIR**.
If input directory contains ".", the files in this directory will be ignored. If input directory is an absolute directory (i.e. "/var/tmp/abc"), it will be stored into the "output-dir/var/tmp/abc". If there are multiple input files or directories, name collision resolution will follow the same rules as **--output-dir-flat**.
- **--format=FORMAT**: compress and decompress in other formats. If compiled with support, zstd can compress to or decompress from other compression algorithm formats. Possibly available options are **zstd**, **gzip**, **xz**, **lzma**, and **lz4**. If no such format is provided, **zstd** is the default.
- **-h/-H, --help**: display help/long help and exit
- **-V, --version**: display version number and exit. Advanced : **-vV** also displays supported formats. **-vvV** also displays POSIX support. **-q** will only display the version number, suitable for machine reading.
- **-v, --verbose**: verbose mode, display more information
- **-q, --quiet**: suppress warnings, interactivity, and notifications. specify twice to suppress errors too.
- **--no-progress**: do not display the progress bar, but keep all other messages.
- **--show-default-cparams**: Shows the default compression parameters that will be used for a particular src file. If the provided src file is not a regular file (eg. named pipe), the cli will just output the default parameters. That is, the parameters that are used when the src size is unknown.
- **--**: All arguments after **--** are treated as files

Restricted usage of Environment Variables

Using environment variables to set parameters has security implications. Therefore, this avenue is intentionally restricted. Only **ZSTD_CLEVEL** and **ZSTD_NBTHREADS** are currently supported. They set the compression level and number of threads to use during compression, respectively.

ZSTD_CLEVEL can be used to set the level between 1 and 19 (the "normal" range). If the value of **ZSTD_CLEVEL** is not a valid integer, it will be ignored with a warning message. **ZSTD_CLEVEL** just replaces the default compression level (3).

ZSTD_NBTHREADS can be used to set the number of threads **zstd** will attempt to use during compression. If the value of **ZSTD_NBTHREADS** is not a valid unsigned integer, it will be ignored with a warning message. **ZSTD_NBTHREADS** has a default value of (1), and is capped at **ZSTDMT_NBWORKERS_MAX==200**. **zstd** must be compiled with multithread support for this to have any effect.

They can both be overridden by corresponding command line arguments: **-#** for compression level and **-T#** for number of compression threads.

Parallel Zstd OPTIONS

Additional options for the pzstd utility

- **-p, --processes**
number of threads to use for (de)compression (default:4)

DICTIONARY BUILDER

zstd offers *dictionary* compression, which greatly improves efficiency on small files and messages. It's possible to train **zstd** with a set of samples, the result of which is saved into a file called a **dictionary**. Then during compression and decompression, reference the same dictionary, using command **-D dictionaryFileName**. Compression of small files similar to the sample set will be greatly improved.

--train FILES

Use FILES as training set to create a dictionary. The training set should contain a lot of small files (> 100), and weight typically 100x the target dictionary size (for example, 10 MB for a 100 KB dictionary).

Supports multithreading if **zstd** is compiled with threading support. Additional parameters can be specified with **--train-fastcover**. The legacy dictionary builder can be accessed with **--train-legacy**. The cover dictionary builder can be accessed with **--train-cover**. Equivalent to **--train-fastcover=d=8,steps=4**.

-o file Dictionary saved into **file** (default name: dictionary).

--maxdict=#

Limit dictionary to specified size (default: 112640).

-# Use # compression level during training (optional). Will generate statistics more tuned for selected compression level, resulting in a *small* compression ratio improvement for this level.

-B# Split input files in blocks of size # (default: no split)

--dictID=#

A dictionary ID is a locally unique ID that a decoder can use to verify it is using the right dictionary. By default, zstd will create a 4-bytes random number ID. It's possible to give a precise number instead. Short numbers have an advantage : an ID < 256 will only need 1 byte in the compressed frame header, and an ID < 65536 will only need 2 bytes. This compares favorably to 4 bytes default. However, it's up to the dictionary manager to not assign twice the same ID to 2 different dictionaries.

---train-cover=[k=#,d=#,steps=#,split=#,shrink=[#]]

Select parameters for the default dictionary builder algorithm named cover. If *d* is not specified, then it tries *d* = 6 and *d* = 8. If *k* is not specified, then it tries *steps* values in the range [50, 2000]. If *steps* is not specified, then the default value of 40 is used. If *split* is not specified or *split* <= 0, then the default value of 100 is used. Requires that *d* <= *k*. If *shrink* flag is not used, then the default value for *shrinkDict* of 0 is used. If *shrink* is not specified, then the default value for *shrinkDictMaxRegression* of 1 is used.

Selects segments of size *k* with highest score to put in the dictionary. The score of a segment is computed by the sum of the frequencies of all the subsegments of size *d*. Generally *d* should be in the range [6, 8], occasionally up to 16, but the algorithm will run faster with *d* <= 8. Good values for *k* vary widely based on the input data, but a safe range is [2 * *d*, 2000]. If *split* is 100, all input samples are used for both training and testing to find optimal *d* and *k* to build dictionary. Supports multithreading if **zstd** is compiled with threading support. Having *shrink* enabled takes a truncated dictionary of minimum size and doubles in size until compression ratio of the truncated dictionary is at most *shrinkDictMaxRegression*% worse than the compression ratio of the largest dictionary.

Examples:

zstd --train-cover FILES

zstd --train-cover=k=50,d=8 FILES

zstd --train-cover=d=8,steps=500 FILES

zstd --train-cover=k=50 FILES

zstd --train-cover=k=50,split=60 FILES

zstd --train-cover=shrink FILES

zstd --train-cover=shrink=2 FILES

---train-fastcover=[k=#,d=#,f=#,steps=#,split=#,accel=#]

Same as cover but with extra parameters *f* and *accel* and different default value of *split*. If *split* is not specified, then it tries *split* = 75. If *f* is not specified, then it tries *f* = 20. Requires that 0 < *f* < 32. If *accel* is not specified, then it tries *accel* = 1. Requires that 0 < *accel* <= 10. Requires that *d* =

6 or $d = 8$.

f is log of size of array that keeps track of frequency of subsegments of size d . The subsegment is hashed to an index in the range $[0, 2^f - 1]$. It is possible that 2 different subsegments are hashed to the same index, and they are considered as the same subsegment when computing frequency. Using a higher f reduces collision but takes longer.

Examples:

zstd --train-fastcover FILEs

zstd --train-fastcover=d=8,f=15,accel=2 FILEs

--train-legacy[=selectivity=#]

Use legacy dictionary builder algorithm with the given dictionary *selectivity* (default: 9). The smaller the *selectivity* value, the denser the dictionary, improving its efficiency but reducing its possible maximum size. **--train-legacy=s=#** is also accepted.

Examples:

zstd --train-legacy FILEs

zstd --train-legacy=selectivity=8 FILEs

BENCHMARK

-b# benchmark file(s) using compression level #

-e# benchmark file(s) using multiple compression levels, from **-b#** to **-e#** (inclusive)

-i# minimum evaluation time, in seconds (default: 3s), benchmark mode only

-B#, --block-size=#

cut file(s) into independent blocks of size # (default: no block)

--priority=rt

set process priority to real-time

Output Format: CompressionLevel#Filename : InputSize -> OutputSize (CompressionRatio), CompressionSpeed, DecompressionSpeed

Methodology: For both compression and decompression speed, the entire input is compressed/decompressed in-memory to measure speed. A run lasts at least 1 sec, so when files are small, they are compressed/decompressed several times per run, in order to improve measurement accuracy.

ADVANCED COMPRESSION OPTIONS

-B#:

Select the size of each compression job. This parameter is only available when multi-threading is enabled. Each compression job is run in parallel, so this value indirectly impacts the nb of active threads. Default job size varies depending on compression level (generally $4 * \text{windowSize}$). **-B#** makes it possible to manually select a custom size. Note that job size must respect a minimum value which is enforced transparently. This minimum is either 512 KB, or **overlapSize**, whichever is largest. Different job sizes will lead to (slightly) different compressed frames.

--zstd[=options]:

zstd provides 22 predefined compression levels. The selected or default predefined compression level can be changed with advanced compression options. The *options* are provided as a comma-separated list. You may specify only the options you want to change and the rest will be taken from the selected or default compression level. The list of available *options*:

strategy=*strat*, **strat=***strat*

Specify a strategy used by a match finder.

There are 9 strategies numbered from 1 to 9, from faster to stronger: 1=ZSTD_fast, 2=ZSTD_dfast, 3=ZSTD_greedy, 4=ZSTD_lazy, 5=ZSTD_lazy2, 6=ZSTD_btlazy2, 7=ZSTD_btopt, 8=ZSTD_btultra, 9=ZSTD_btultra2.

windowLog=wlog, wlog=wlog

Specify the maximum number of bits for a match distance.

The higher number of increases the chance to find a match which usually improves compression ratio. It also increases memory requirements for the compressor and decompressor. The minimum *wlog* is 10 (1 KiB) and the maximum is 30 (1 GiB) on 32-bit platforms and 31 (2 GiB) on 64-bit platforms.

Note: If **windowLog** is set to larger than 27, **--long=windowLog** or **--memory=windowSize** needs to be passed to the decompressor.

hashLog=hlog, hlog=hlog

Specify the maximum number of bits for a hash table.

Bigger hash tables cause less collisions which usually makes compression faster, but requires more memory during compression.

The minimum *hlog* is 6 (64 B) and the maximum is 30 (1 GiB).

chainLog=clog, clog=clog

Specify the maximum number of bits for a hash chain or a binary tree.

Higher numbers of bits increases the chance to find a match which usually improves compression ratio. It also slows down compression speed and increases memory requirements for compression. This option is ignored for the ZSTD_fast strategy.

The minimum *clog* is 6 (64 B) and the maximum is 29 (524 Mib) on 32-bit platforms and 30 (1 Gib) on 64-bit platforms.

searchLog=slog, slog=slog

Specify the maximum number of searches in a hash chain or a binary tree using logarithmic scale.

More searches increases the chance to find a match which usually increases compression ratio but decreases compression speed.

The minimum *slog* is 1 and the maximum is 'windowLog' - 1.

minMatch=mml, mml=mml

Specify the minimum searched length of a match in a hash table.

Larger search lengths usually decrease compression ratio but improve decompression speed.

The minimum *mml* is 3 and the maximum is 7.

targetLength=tlen, tlen=tlen

The impact of this field vary depending on selected strategy.

For ZSTD_btopt, ZSTD_btultra and ZSTD_btultra2, it specifies the minimum match length that causes match finder to stop searching. A larger **targetLength** usually improves compression ratio but decreases compression speed. For ZSTD_fast, it triggers ultra-fast mode when > 0. The value represents the amount of data skipped between match sampling. Impact is reversed : a larger **targetLength** increases compression speed but decreases compression ratio.

For all other strategies, this field has no impact.

The minimum *tlen* is 0 and the maximum is 128 Kib.

overlapLog=ovlog, ovlog=ovlog

Determine **overlapSize**, amount of data reloaded from previous job. This parameter is only available when multithreading is enabled. Reloading more data improves compression ratio, but decreases speed.

The minimum *ovlog* is 0, and the maximum is 9. 1 means "no overlap", hence completely independent jobs. 9 means "full overlap", meaning up to **windowSize** is reloaded from previous job. Reducing *ovlog* by 1 reduces the reloaded amount by a factor 2. For example, 8 means "windowSize/2", and 6 means "windowSize/8". Value 0 is special and means "default" : *ovlog* is

automatically determined by **zstd**. In which case, *ovlog* will range from 6 to 9, depending on selected *strat*.

IdmHashLog=*lhlog*, lhlog=*lhlog*

Specify the maximum size for a hash table used for long distance matching.

This option is ignored unless long distance matching is enabled.

Bigger hash tables usually improve compression ratio at the expense of more memory during compression and a decrease in compression speed.

The minimum *lhlog* is 6 and the maximum is 30 (default: 20).

IdmMinMatch=*lmml*, lmml=*lmml*

Specify the minimum searched length of a match for long distance matching.

This option is ignored unless long distance matching is enabled.

Larger/very small values usually decrease compression ratio.

The minimum *lmml* is 4 and the maximum is 4096 (default: 64).

IdmBucketSizeLog=*lblog*, lblog=*lblog*

Specify the size of each bucket for the hash table used for long distance matching.

This option is ignored unless long distance matching is enabled.

Larger bucket sizes improve collision resolution but decrease compression speed.

The minimum *lblog* is 1 and the maximum is 8 (default: 3).

IdmHashRateLog=*lhrlog*, lhrlog=*lhrlog*

Specify the frequency of inserting entries into the long distance matching hash table.

This option is ignored unless long distance matching is enabled.

Larger values will improve compression speed. Deviating far from the default value will likely result in a decrease in compression ratio.

The default value is **wlog – lhlog**.

Example

The following parameters sets advanced compression options to something similar to predefined level 19 for files bigger than 256 KB:

--zstd=wlog=23,clog=23,hlog=22,slog=6,mml=3,tlen=48,strat=6

BUGS

Report bugs at: <https://github.com/facebook/zstd/issues>

AUTHOR

Yann Collet

NAME

edrcrevision – revision history of WA2L/edrc

AVAILABILITY

WA2L/edrc

DESCRIPTION

This manpage gives you an overview of changes between the different releases of WA2L/edrc.

For an explanation of the revision number system see section **RELEASE-NUMBERS** and for an explanation of used shortcuts in the **REVISION-HISTORY** see section **SHORTCUTS**.

RELEASE-NUMBERS

The release number format is: <major>.<update>.<patch_level> .

If the <major> part changes, major design changes were applied, therefore you have to upgrade your installation.

If the <update> part changes, important bugfixes were applied, the config file might have changed, therefore you should upgrade your WA2L/edrc installation to the current version.

If the <patch_level> part changes, minor changes or bugfixes were applied, the configuration file format did not change, therefore it is not urgent to upgrade your installation, but recommended.

SHORTCUTS**OBJECT**

| | |
|------------|---|
| cmd | command |
| man | manual page |
| doc | plain documentation |
| cfg | configuration |
| opt | option |
| dir | directory |
| lib | shared or static operating system dependent library files or perl modules |

ACTION

new new function, configuration parameter, option, behavior

chg change

fix bugfix

cor correction of descriptions, comments, logfile output, typing errors

REVISION-HISTORY

1.5.57

Sun Dec 22 22:22:22 CEST 2024 release

MAIN IMPROVEMENTS:

- obj/act *: several minor corrections and improvements.
- cmd/new lgcheckd: more efficient and leaner log check daemon to replace the logcheckd.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.57 (Sat May 25 20:19:00 CEST 2024)
- cfg/chg syslog.Linux-64: significant enhancement of pattern file.
- cfg/chg ..name_index: . to document the current directory added in many directories.
- cfg/chg edrcman.map: many more man pages and elements mapped to manual pages to enable looking up as most files as possible.
- cfg/chg logcheckd.cfg: new setting USE_LEGACY_COMMAND to transparently relay to new log check daemon lgcheckd.
- cfg/chg edrcinit.cfg: new setting nginx_shellinaboxd.
- cfg/chg edrcinit.cfg: new setting batteryalert.
- cfg/chg binprobe.dat: add md5string.
- cfg/new edrcinit/WA2Ledrc.batteryalert: edrcinit service handler for batteryalert.
- cmd/chg vsav: save file versions compressed.
- cmd/chg edrcman: the OS in -o OS manpage is not case sensitive.
- cmd/chg pack: react on \$EDRC_DEBUG_PACK_EXCLUDE to debug the EXCLUDE_LIST of the pack.cfg file.
- cmd/chg pack: add the pack.cfg file at the beginning of the package file to be more error resilient.
- cmd/chg patchinstall: repeat file installation and symlink property adjustment after corpse removal on cpio extract error.
- cmd/chg lua, luac: moved from edrc/lib/ to edrc/bin/.
- cmd/chg job, joblog, jobwatch, joblog: new option -V to print program version.
- cmd/chg job, jobwatch: state in job list is colored on terminal.
- cmd/chg perlenv: new -n option to avoid export VARIABLE output.
- cmd/chg shlib: new -n option to avoid export VARIABLE output.
- cmd/chg joblog: new -q option to tail always the current running job log.
- cmd/chg www/report/cgi/vsdfml: display vsdfml files in WA2L/edrc:report portal.
- cmd/chg www/report/cgi/md: display markdown files in WA2L/edrc:report portal.
- cmd/chg www/report/cgi/man: display man files in WA2L/edrc:report portal.
- cmd/chg www/report/cgi/edoc: display files formatted with contrib.doc syntax in WA2L/edrc:report portal.
- cmd/chg www/report/cgi/doc: enhancement to display vsdfml, md, man, doc and compressed files in WA2L/edrc:report portal.
- cmd/chg select_columns: enable to display CSV files with an initial sep=... line properly.
- cmd/chg lgcpattern: add -i option to only print [IGNORED] lines.
- cmd/chg lgcpattern: add -p option to print a regular expression based on logfile entries.
- cmd/chg batteryalert: add -a option to query alert state.
- cmd/chg csvcat: enable to display CSV files with an initial sep=... line properly.
- cmd/new cpanm: get, unpack build and install modules from CPAN.
- cmd/new luarocks: Lua package installation program.
- cmd/new luaversion: print lua version.
- cmd/new .lua_wrapper: lua wrapper.

- cmd/new luaenv: print environment for lua.
- cmd/new pip: Python package manager to handle python packages distributed with WA2L/edrc.
- cmd/new pythonversion: print major.minor python version.
- cmd/new name: handle also files started thru .python_wrapper.
- cmd/chg edrcman: add -i option to display internet based man pages.
- cmd/chg edrcman: add/chg -l (sys|os) option to list available local and internet based man page sources.
- cmd/chg cmdlist: handle also python scripts started thru .python_wrapper.
- cmd/chg .python_wrapper: wrap python commands.
- cmd/new pythonenv: print environment needed for external python commands or recovery scripts written in python to profit from WA2L/edrc distributed packages.
- cmd/chg contrib/scriptinstall: avoid error message on empty scripts dir. cmd/chg csv: skip empty and commented lines at input begin for w2s, t2s, ct2 commands.
- cmd/chg vls: list last (terminal page) of saved versions, whereas vlist lists all saved versions (unchanged).
- cmd/chg name: avoid read permission error messages.
- cmd/chg name: read '.' entry in directory/..name_index to print directory description.
- cmd/chg tpl: \$TPL_CFG can hold an additional configuration file for tpl.
- cmd/chg www/report/cgi/statistics: more accurate image alignment.
- cmd/chg www/report/cgi/users.list: timestamp format corrected.
- cmd/chg www/report/cgi/users.add: timestamp format corrected.
- cmd/chg www/report/cgi/users.*: use post in forms instead of get.
- cmd/chg sqlite: upgrade to version 3.46.0 for Linux.64/2.27/.
- cmd/chg sqlite: upgrade to version 3.46.0 for Linux.64/2.22/.
- cmd/chg job: last targets the last job enqueued by the user, now.
- cmd/chg pmdesc: new option -d for csv output, set default width of module name output to 52 characters.
- cmd/new lgcheckd: more efficient and leaner log check daemon to replace the logcheckd.
- cmd/chg csv: option cat:lgcheckd.cfg added.
- cmd/chg pstree: use long output of process list on Linux and limit output width on a TTY. Non-TTY output is not limited.
- cmd/chg jobstart: new option logswitch to switch logfile.
- cmd/chg pstree: exit code 5 on abort.
- cmd/chg pstree: new option -r to print process-ids in reverse order.
- cmd/chg logview: new option -h to print program usage.
- cmd/chg logtail: new option -h to print program usage, -c and -C to colorize tailed output based on pattern files.
- cmd/chg logtail: new option -t to add a high resolution timestamp to the output.
- cmd/chg jobstart: add DOCREF column to jobstart list output.
- cmd/new batteryalert: monitor battery charge level and alert.
- cmd/chg job: -c option to print commands fo job.
- cmd/chg job: remove job improved.
- cmd/chg lgcpattern: check a log check pattern file definition.
- cmd/new eterm: start an application in a separate X-terminal.
- cmd/chg job: state change (run/done/stab/abort/removed) are now also logged to the log file of the job to have the information also available when using joblog -j jobid.
- cmd/chg edrcinit: new options cfg-([un])(activate|guard)) to set the related configuration settings in the configuration file thru the command line.
- cmd/new md5string: calculate MD5 checksum for strings line by line.
- cmd/chg job: FUNCTIONs to be used in jobs: JOBBANNER, JOBTITLE, JOBSUBTITLE, JOBSUBSUBTITLE, JOBTAG, JOBCOMPLETED.
- cmd/chg job: -a option to print job run statistics for accounting purposes.
- cmd/chg job: print TAG when using job -p jobid.
- cmd/chg jobstart: cache file timestamp is no longer changed if no related logfile is changed.
- cmd/chg jobstart: avoid error message if no cache file was created yet.
- cmd/chg logviewer: increase resilience to filesystem shortage on report generation.

- cmd/chg name: enable handling files and directories with spaces in the file- or directory-name.
- cmd/chg cfg2html: new version 7.1.2.1 for Linux and Linux-64.
- cmd/chg lscomp: handle .whl, .rock files.
- cmd/chg llcomp: handle .whl, .rock files.
- cmd/chg catcomp: handle .whl, .rock files.
- man/chg *: *: formatting correction of many manual pages.
- man/chg pack.1m: description of \$EDRC_DEBUG_PACK_EXCLUDE to debug the EXCLUDE_LIST of the pack.cfg file.
- man/chg perlenv.3: description of -n option to avoid export output.
- man/chg shlib.3: description of -n option to avoid export output.
- man/chg joblog.1: description of -q option to tail always the current run log file.
- man/chg pkgdir.4: replace man command call with edrcman command call.
- man/chg pkgdir.4: add PIP package handling.
- man/chg tpl.1: environment variable \$TPL_CFG description.
- man/chg shell.cfg.4: environment variable \$TPL_CFG description.
- man/chg utility.h.3: synopsis corrected.
- man/chg strings.h.3: include file now located in edrc/lib/\$OSID/includes/.
- man/chg utility.h.3: include file now located in edrc/lib/\$OSID/includes/.
- man/chg program.h.3: include file now located in edrc/lib/\$OSID/includes/.
- man/chg checkopt.h.3: include file now located in edrc/lib/\$OSID/includes/.
- man/chg job.1: adjust description of last, add .job.last file description.
- man/chg pmdesc.1: description of -d option added, manual page formatting changed.
- man/new cpanm.1
- man/new luarocks.1
- man/new luarocks.cfg.4
- man/new luaenv.3
- man/new luaversion.3
- man/new lua_wrapper.1
- man/new lua_wrapper.cfg.4
- man/new lgcheckd.1m
- man/new lgcheckd.cfg.4
- man/new logcheckd.pattern.4: adjustments to cover also lgcheckd.
- man/new logcheckd.interface.4: new environment variables \$APPROOT, \$SERVER_ENVIRONMENT_NAME, \$SERVER_ENVIRONMENT_CUSTOMER, \$SERVER_ENVIRONMENT_DESCRIPTION that can be used in interfaces.
- man/new logcheckd.interface.3: adjustments to cover also lgcheckd.
- man/new logcheckd.1m: hint to lgcheckd.
- man/new lgcheckd.state.db.4: description of the state.db structure.
- man/new python_wrapper.1
- man/new pythonversion.3
- man/new pythonenv.3
- man/chg csv.3: description of csv cat:lgcheckd.cfg option.
- man/chg jobstart.1: description of logswitch option.
- man/chg pstree.1: description of exit code 5.
- man/chg pstree.1: description of -r option to print process-ids in reverse order.
- man/chg logview.1: description of -h option to print the program usage.
- man/chg logtail.1: description of -h option to print the program usage, -c and -C to colorize output.
- man/chg logtail.1: description of -t to add a high resolution to the output.
- man/chg jobstart.1m: description of DOCREF column in jobstart list output.
- man/new batteryalert.1: description of batteryalert command.
- man/chg job.1: description of -c option.
- man/new lgcpattern.3
- man/new wa2ledrc_shellinaboxd.1

- man/chg EDRC.1: add eterm.1, pythonenv.1, python_wrapper.1, pip.1.
- man/chg edrcintro.1: add eterm.1.
- man/new eterm.1
- man/chg edrcinit.1m: description of new options cfg-([un](activate|guard)).
- man/new md5string.1
- man/chg job.1: description of -a option and job FUNCTIONS.
- man/chg lspm.1: add SEE ALSO section.
- man/new pip.1
- man/new pip3.1
- man/new pip3-check.1
- man/new pip3-wheel.1
- man/new pip3-uninstall.1
- man/new pip3-show.1
- man/new pip3-search.1
- man/new pip3-list.1
- man/new pip3-install.1
- man/new pip3-help.1
- man/new pip3-hash.1
- man/new pip3-freeze.1
- man/new pip3-download.1
- doc/cor WA2Ledrc:report/help: typing errors.
- doc/chg www/report/html/help.html: description of handling of certain file extensions added
- dir/new lib/python/: storage for python scripts within WA2L/edrc started thru .python_wrapper.
- dir/new lib/python/pym/: location of python modules in a virtual environment (venv) within WA2L/edrc started thru .python_wrapper.

1.5.56

Fri May 24 22:08:13 CEST 2024 release

MAIN IMPROVEMENTS:

- cmd/chg tpl: select and print template file.
- cmd/chg csv: many improvements.
- cmd/chg hostlist: allow output of Ansible inventory in ini-, json- and yaml file format.
- cmd/chg www/foswiki: Foswiki is no longer distributed with WA2L/edrc.
- cmd/new jobstart: run jobs thru a simple interface.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.56 (Thu Oct 21 21:42:36 METDST 2021)
- cfg/chg edrc*.cfg: fallback to linux -> vt220, if TERM=terminal does not exist.
- cfg/chg job.cfg: new setting COLORIZE to switch off colorizing output on terminal.
- cfg/chg binprobe.dat: special probe command for bzmre on Ubuntu, more descriptive output of special probe commands on certain preconditions.
- cmd/chg tpl: select and print template file.
- cmd/chg csv: new option variable to print a record as initialize-able variables.
- cmd/chg csv: new option row to return a row by linenumber.
- cmd/chg csv: new option wherenot and wherenot:ignorecase for inverse data select.
- cmd/chg csv: new option where:ignorecase for non case sensitive data select.
- cmd/chg csv: new option cat:authorized_keys for .ssh/authorized_keys files.
- cmd/chg csv: new option cat:known_hosts for .ssh/known_hosts files.
- cmd/chg csv: new option comm, comm:both, comm:stdin, comm:file to compare CSV data.
- cmd/chg csv: new option addcol to add a column.
- cmd/chg csv: new option order to reorder the columns.
- cmd/chg csv: new option transpose to transpose a csv file.
- cmd/chg csv: new option renamecol to rename columns.
- cmd/chg csv: new option sed to sed on data rows.
- cmd/chg csv: new option t2s to convert tab separations to semicolon separations.

- cmd/chg csv: new option w2s to convert white-space separations to semicolon separations.
- cmd/chg csv: new option avtar to create CSV from Avamar avtar output.
- cmd/chg csv: new option mccli to create CSV from Avamar mccli output.
- cmd/chg csv: new option cat:list to create CSV from csv list output.
- cmd/chg csv: new option diff to diff CSV stream with file.
- cmd/chg csv: new option unselect to de-select columns.
- cmd/chg csv: new option list:nocount to for list output without counter line.
- cmd/chg csv: new option list:*.nocount to for list output without counter line.
- cmd/chg csv: new option cat:mailcap for '/etc/mailcap' format.
- cmd/chg contrib/scriptdocbook: create a single-file menu (accessible thru index.htm) to enable serving a ScriptDocBook from servers blocking JavaScript files.
- cmd/chg shell: if WA2L/edrc is not installed on remote system a normal ssh connection without shell is established.
- cmd/new contrib/find: search text in recovery scripts header and documentation.
- cmd/fix edrcman: man compute MAN_PATH as ordered in edrcman.cfg file setting.
- cmd/chg edrcman: possibility to query in bundled OS native manual pages.
- cmd/chg www/report/cgi/doc: correct size listing of symlinked files.
- cmd/new contrib/scriptdocbook: in PATH= of the scriptdocbook.bib file now also files with spaces are possible and when the path begins with "/" it is relative to the recovery script tree.
- cmd/chg hostlist: allow output of Ansible inventory in ini-, json- and yaml file format.
- cmd/fix contrib/search: syntax error on search in scriptcode eliminated.
- cmd/chg httpd: compiled for GLIBC 2.31 (ubuntu).
- cmd/chg WA2L::Util: new function checkopt() to parse options in Perl programs.
- cmd/chg www/foswiki: Foswiki is no longer distributed with WA2L/edrc. The last full distribution which contains Foswiki (version 1.1.8) is WA2L/edrc version 1.5.55. However, it is not actively removed to prevent data loss. To cleanup Foswiki files, remove by hand: edrc/var/www/foswiki/, edrc/etc/*.foswiki.cfg, edrc/lib/edrcinit/WA2Ledrc.foswiki and eliminate the foswiki entries in edrc/etc/edrcinit.cfg.
- cmd/chg remote_shell: new option -N to prevent 'su' to the user given in -s username prior to the connection.
- cmd/chg edrc: the \$EDRC_SCRIPTS_BASEDIR variable is now also available in the file editor and viewer.
- cmd/chg edrc: 'getcwd: cannot access parent dir' error corrected when invoking quit command after distribute -> full.
- cmd/chg tpl: reads also the \$EDRC_SCRIPTS_BASEDIR/tpl.cfg file.
- cmd/chg loggrep: argument parsing improved.
- cmd/chg loggrep: now th first 211 lines are probed for timestamps.
- cmd/chg loggrep, lgrep: performance improvements (on long files).
- cmd/chg loggrep, lgrep: new -NUM option to only analyse the last NUM lines instead of the whole file.
- cmd/chg loggrep, lgrep, logcat, lcat: output when using -M option goes now to fd3 and no longer to stderr.
- cmd/chg binprobe: internal improvements.
- cmd/chg shell: does not any more loose the tty resizing information when connecting to a remote system. Therefore using vi on remote systems is more convenient, now.
- cmd/fix csvcat: no longer remove whitespaces in fields when not before or after field separator.
- cmd/new nc: version 1.89 for Linux/glibc 2.22.
- cmd/chg filedist: print ego message also to stderr to allow better command embedding.
- cmd/new screen: version 4.0.4 for Linux/glibc 2.22.
- cmd/new screen: version 4.0.3 for Linux/glibc 2.27.
- cmd/chg shell: set terminal title when using shell hostname.
- cmd/new purgetemp: purge temporary directories.
- cmd/chg name: now also lists directories that are listed in the ..name_index file.
- cmd/chg logcat: cat log files with time stamps.
- cmd/chg name: also process files with spaces in filename.

- cmd/chg revision: also process files with spaces in filename.
- cmd/chg remote_shell: exit code of ssh/rsh is passed to caller.
- cmd/chg remote_copy: exit code of scp/rcp is passed to caller.
- cmd/chg remote_copy: new option -N to omit su to connection user.
- cmd/chg name: avoid shell interpreting wildcards in file description.
- cmd/new csvq: query csv files with SQL for Linux and Linux-64.
- cmd/chg remote_copy: can now also handle files with spaces.
- cmd/chg remote_copy: new option -v for verbose output.
- cmd/chg contrib/pwsafe: reduce multiple white spaces to a space on output.
- cmd/new uuencode: encode files, compiled for Linux-64/GLIBC 2.27.
- cmd/new decode: decode uuencoded files, compiled for Linux-64/GLIBC 2.27.
- cmd/chg h2: version 2.1.214.
- cmd/chg csv2worksheet: version 2.0.09 use the new POI Java library poi-poi-5.2.2.
- cmd/chg csv2worksheet: very large worksheet support.
- cmd/chg csv2worksheet: support of old *.xls files discontinued.
- cmd/new fit: alias for fit2width.
- cmd/new contrib/scriptinstall: install a script tree package generated using the scriptpack contributed command.
- cmd/new lua: lua language interpreter, compiled for Linux-64, glibc 2.27.
- cmd/new luac: lua language compiler, compiled for Linux-64, glibc 2.27.
- cmd/chg logview: improvement in program abort handling.
- cmd/chg logtail: improvement in program abort handling.
- cmd/chg edrcman: check if man command exists on the system.
- cmd/chg pscount: allow reading list of processes also from file.
- cmd/chg pscount: Performance improvement.
- cmd/chg pscount: no longer react on PRINT_FIT2WIDTH environment variable.
- cmd/chg pscount: spaces at the end of list outputs are eliminated, now.
- cmd/chg pslist: remove additional pslist processes from process list.
- cmd/chg pslist: support very long process lists, now.
- cmd/chg sys2html: create 'System Documentation' HTML reports of remote systems.
- cmd/chg vsav: versioned file save.
- cmd/chg vmore: more saved file version.
- cmd/chg vcat: cat saved file version.
- cmd/chg vgrep: grep pattern in saved file version.
- cmd/chg vdiff: diff to saved file version.
- cmd/chg vlist: list saved file versions.
- cmd/chg vls: list saved file versions.
- cmd/chg vrestore: restore saved file version.
- cmd/new locate: query locate database.
- cmd/new updatedb: update locate database.
- cmd/chg loggrep/logcat: new option -s num to set the number of lines used to resolve the timestamp.
- cmd/chg loggrep/logcat: new option -T id to set the used timestamp id manually.
- cmd/chg stat: new option -L to follow symlinks.
- cmd/chg contrib/scriptpack: extra packed files are now saved to .sav/scriptpack/ of the generated script tree package.
- cmd/new crond: cron daemon.
- cmd/new ecrontab: edit crontab entries for crond.
- cmd/new ecrontab: show next scheduled jobs of crond.
- cmd/new gs: Ghostscript.
- cmd/new fnmatch: match a stream with a wildcard.
- cmd/chg scriptgrep: internally now egrep is used to allow extended regular expressions.
- cmd/chg scriptgrep: matches are colored red on Linux.
- cmd/chg abc, abcd: output of \$e, \$f, \$g, \$TODAY added.

- cmd/chg e, f, g: setting of additional working variables \$e, \$f, \$g.
- cmd/new cde, cdf, cdg: jump to working variables dirs \$e, \$f, \$g.
- cmd/chg pstree: no longer limit output to console width.
- cmd/chg pstree: now also accepts a process name as start point of the process tree.
- cmd/new jobstart: run jobs thru a simple interface.
- cmd/chg timer: date and time options to calculate seconds since epoch for
- cmd/new ps2pdf: PostScript to PDF file conversion.
- cmd/chg .daemon_wrapper: change of CWD to root can be avoided when PWDPRESERVE=daemon_command is set in the daemon_command.
- cmd/chg job: preserve current working directory.
- cmd/chg joblog: now also a log file of a job can be monitored.
- cmd/chg edrcman: output of -l extended by number of man PAGES per operating system.
- cmd/chg job/jobwatch/joblog: now reads the config file job.cfg where the SPOOLBASEDIR can be set.
- cmd/chg .os_wrapper: binary resolution improved. On Linux on missing files sometimes directories with executable permissions set were selected.
- cmd/chg edrcman: more portable checking if man command is installed on the operating system.
- cmd/new lib/man: the man command from mandoc is bundled to provide a man command (called by edrcman and edrc->doc) on systems where the man command is missing.
- cmd/chg woist: do not search a directory twice for a command.
- cmd/chg manvi: uses vsav instead of sav prior to file editing.
- cmd/chg llcomp: support for cpio.xz files added.
- cmd/chg lscomp: support for cpio.xz files added.
- cmd/chg catcomp: support for cpio.xz files added.
- cmd/chg WA2L/edrc:report: statistics simplified and performance increase.
- cmd/chg job: representation of enqueued jobs without observation.
- cmd/chg csv: performance improvement in 'csv header'.
- cmd/chg print_index: now also non-fixed column width is possible.
- cmd/fix edrcinit: services stopped using 'edrcinit stop' are no longer guarded and remain stopped until 'edrcinit start [name]' is invoked.
- cmd/chg edrcinit: new option checknow to force an immediate service check.
- cmd/chg edrcinit: when 'edrcinit unguard' is used, the services remain unguarded until 'edrcinit guard' is used. Therefore the service guarding is no longer influenced by other options.
- cmd/new zstd: compress and uncompress files.
- cmd/new unzstd: uncompress files.
- cmd/new unczst: uncompress zstd-compressed cpio files.
- cmd/new unczstd: uncompress zstd-compressed cpio files.
- cmd/new untzst: uncompress zstd-compressed tar files.
- cmd/new untzstd: uncompress zstd-compressed tar files.
- cmd/chg catcomp can now also handle zstd-compressed cpio and tar files.
- cmd/chg lscomp can now also handle zstd-compressed cpio and tar files.
- cmd/chg llcomp can now also handle zstd-compressed cpio and tar files.
- cmd/chg unrpm can now also handle rpm packages that contain a .cpio.zst archive.
- cmd/chg cfg2html: new version 7.0.1.1 for Linux and Linux-64.
- cmd/chg tomorrow, yesterday: more stable return value.
- cmd/new tscat: filter to add a timestamp and filename to a stream.
- cmd/chg job: enable -f to force to stab a running job with running processes.
- cmd/chg job: add DURATION to output of job -p.
- cmd/fix pstree: no process output when given process id does not exist.
- cmd/fix pstree: message prefix corrected to pstree.
- cmd/new md2html: convert markdown file to html.
- cmd/chg print_index: new option to eliminate trailing spaces in output.
- cmd/chg cmdlist: startmode direct/daemon added, output of commands started thru the .os_wrapper improved.

- cmd/chg revision: now where possible revision of target file is listed for wrapped commands (not the revision of the wrapper).
- cmd/chg name: now where possible name of target file is listed for wrapped commands (not the name of the wrapper).
- cmd/new pod2html: convert POD to html.
- cmd/new cdrpt: cd var/www/report/rpt.
- cmd/new kalc: RPN calculator.
- cmd/new contrib/kalc: RPN calculator as contributed command in edrc.
- cmd/chg rssh: -l option to list and select from past connections.
- cmd/chg patchinstall: -i option to list installed patches.
- cmd/new contrib/bc: bc calculator as contributed command in edrc.
- cmd/new vvi: versioned file editor.
- dir/new var/samples/templates/: templates to be used with the tpl command.
- dir/new var/vsav/files/: storage for vsav.
- man/new tpl.1
- man/new tpl.cfg.1
- man/chg job.cfg.4.1: description of COLORIZE setting to control coloring job list output.
- man/chg compatibility.1: add tpl.
- man/chg compatibility.1: add YSCR for python scripts.
- man/chg edrcports.4: add SMTP, enhance port requirement documentation.
- man/chg csv.3: description of wherenot, wherenot:ignorecase, where:ignorecase, cat:authorized_keys, and cat:known_hosts option added.
- man/new ascii.3
- man/new charsets.3
- man/new utf-8.3
- man/chg shell: description of host connection when WA2L/edrc is not installed.
- man/chg shell: EXIT STATUS descriptions enhanced.
- man/chg manpages.4: description to .PP to end indent in list.
- man/chg edrcman.1: add description of man/OS/<OSID>/ directory.
- man/new OpenSUSE, Ubuntu, OpenSolaris: native manual pages bundled with WA2L/edrc.
- man/chg contrib.edrc.1m: description of the changed PATH= settings.
- man/chg hostlist.3: description of the Ansible inventory ini- and yaml file format output options.
- man/new wa2ledrc_edrcapi.1: description of WA2L/edrc:edrcapi.
- man/chg wa2ledrc_report.1: add documentation of https using nginx.
- man/chg wa2l_util.3: documentation of checkopt() function.
- man/chg edrcports.4: add edrcapi.
- man/chg remote_shell.1: documentation of the new -N option.
- man/chg tpl.1: documentation of \$EDRC_SCRIPTS_BASEDIR/tpl.cfg.
- man/chg loggrep.1: description of -NUM option.
- man/chg loggrep.1: description of -M option which output goes to fd3 now.
- man/chg remote_shell.3: description of exit code 1##.
- man/chg remote_copy.3: description of exit code 1##.
- man/chg remote_copy.3: description of -N option.
- man/new TLDP: version 5.13 of TLDP (The Linux Documentation Project) manual pages included.
- man/new csvq.3: short description for csvq.
- man/chg remote_copy.3: description of -v option.
- man/new uuencode.1
- man/new uuencode.4
- man/new uuencode.1
- man/new lua.3
- man/new luac.3
- man/chg pscount.3: description of -f option.
- man/chg pscount.3: \$PRINT_FIT2WIDTH removed.

- man/new sys2html.1m
- man/chg edrcintro.1: sys2html added
- man/new vsav.1: manual page for vsav, vcat, vmore, vdiff, vlist, vrestore.
- man/new locate.1
- man/new updatedb.1
- man/new locatedb.4
- man/chg contrib.edrc.1m: description of the directory where extra files are packed to using scriptpack.
- man/new crond.3
- man/new ecrontab.1
- man/new ecrontab.4
- man/new gs.3
- man/new fnmatch.3
- man/chg pstree.1: description of processname.
- man/new jobstart.1
- man/new ps2pdf.3
- man/chg daemon_wrapper.1: documentation of PWDPRESERVE=command.
- man/chg joblog.1: documentation of new -j id option.
- man/new job.cfg.4
- man/chg llcomp.1: support for cpio.xz.
- man/chg lscomp.1: support for cpio.xz.
- man/chg catcomp.1: support for cpio.xz.
- man/chg edrcinit.1m: improved.
- man/new pkgdir.4: directory of package/software handling commands.
- man/new tscat.3
- man/new md2html.3
- man/chg print_index.3: description of new nospace option.
- man/chg cmdlist.1m: description of daemon start mode added.
- man/chg job.1: purpose description of the job state files.
- man/new pod2html.3
- man/new perlpod.4
- man/new kalc.1
- man/chg rssh.1: -l option to list and select from past connections.
- man/chg patchinstall.1m: -l option to list installed patches.
- man/new edrclicense.4: GNU General Public License for WA2L/edrc.
- man/new vvi.1: versioned file edit.
- doc/chg WA2Ledrc:report/help: provide access to local- and sourceforge-manuals.
- doc/chg WA2Ledrc:report/doc: new DESCRIPTION column for report sub-folders whereas the description is also provided thru a ..du_index file owned by edrc in the related folder.

1.5.55

Sun Oct 10 22:23:55 METDST 2021 release

MAIN IMPROVEMENTS:

- cmd/chg tinydns, tinydns-data, dnscache, djbdns: removed from WA2L/edrc because it does not support IPV4 and IPV6, probably YADIFA will be integrated.
- cmd/new rosid: resolve osid of a remote system.
- cmd/new rcat: display (cat) remote- or local file(s).
- cmd/new rdif: display differences (diff) of remote- or local file(s).
- cmd/new rcomm: compare (comm) remote- or local file(s).
- cmd/new loggrep: (aka. lgrep) grep selected time range in log files.
- cmd/new contrib/clock: a clock in military format.
- cmd/new contrib/search: search text in recovery scripts.
- cmd/chg exiftool: upgraded to version 12.26.
- man/new many

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.55 (Sat Nov 14 19:11:54 MET 2020)
- cfg/chg www/report/.meta/VERSION: 1.0.20.
- cfg/new lynx.lss: color configuration file for lynx.
- cfg/chg kshrc: set ls colors for .xcf.
- cfg/chg kshrc: save directory stack on shell exit to enable ret on re-entry.
- cmd/chg whoisin: now uses ssh-exec internally and no longer rlogin.
- cmd/new wmic: for Linux-64, running on GLIBC 2.27.
- cmd/new wmic: for Linux-64, running on GLIBC 2.8.
- cmd/new winexe: for Linux-64, running on GLIBC 2.8.
- cmd/chg tinydns, tinydns-data, dnscache, djbdns: removed from WA2L/edrc because it does not support IPV4 and IPV6, probably YADIFA will be integrated.
- cmd/new scriptmenupath: return script menu path in form 'menu -> submenu -> subsubmenu -> menu-point'.
- cmd/new rosid: resolve osid of a remote system.
- cmd/chg .os_wrapper: more save internal glibc.version resolution.
- cmd/new contrib/clock: a clock in military format.
- cmd/chg scriptsequence: log to var/log/scriptsequence.log.
- cmd/new csv: filter to process CSV data.
- cmd/chg contrib/doc: avoid code output on first line before .SS.
- cmd/chg edrc: WARNING on removed menu when invoking a contributed command.
- cmd/chg ssh-exec: mode of executing commands changed to avoid problems with quotes.
- cmd/chg ssh-exec: it is now possible to start the command as user different from the START_USER setting in ssh-exec.cfg as long as the local user specified in the -l option is equal to the user's ID.
- cmd/chg ssh-exec: -b option to avoid remote banner output.
- cmd/fix ssh-exec: usage parsing.
- cmd/chg www/report/js/report.js: formatting of the 'leaving web server message' changed.
- cmd/chg lynx: version 2.8.9rel.1 (glibc 2.17) compiled with ssl (HTTPS) and bundled for Linux-64.
- cmd/chg lynx: version 2.8.9rel.1 (glibc 2.8) compiled with ssl (HTTPS) and bundled for Linux-64.
- cmd/new shell: grep is an alias for grep --color on *Linux* to colorize matching patterns.
- cmd/new shell: egrep is an alias for egrep --color on *Linux* to colorize matching patterns.
- cmd/chg shell: prints README contents to stderr when doing a 'cd' to a directory where a README file exists.
- cmd/chg scriptextract: no longer use the portable cpio format (-c switch) to extract the distribution file for Linux-64.
- cmd/new rcat: display (cat) remote- or local file(s).
- cmd/new rdiff: display differences (diff) of remote- or local file(s).
- cmd/new rcomm: compare (comm) remote- or local file(s).
- cmd/chg contrib/logviewer: can now include additional extra log files.
- cmd/new loggrep: grep selected time range in log files.
- cmd/new textblock: format a stream into a justified text block with an optional hanging first line.
- cmd/chg xmore: now catches Ctrl+C and cleanly aborts to avoid a mangled terminal. This had mostly an impact when using vsdfml in a recovery script and the user pressed Ctrl+C while still viewing the file.
- cmd/new ansi2txt: convert ANSI color codes to text.
- cmd/new ansi2txt: compiled for Linux/GLIBC 2.8.
- cmd/new contrib/search: search text in recovery scripts.
- cmd/chg tcpdump: version 3.9.9 (GLIBC 2.8) bundled for Linux-64.
- cmd/chg edrcman: on SuSE 12.3 and 12.5 with man versions 2.63 or 2.6.6 manual pages residing in directories having a dash in it could not be found. The edrcman command now is resistant to this and regardless can displays the manual page.
- cmd/fix contrib/doc: do not compute '-' and '/' in O: sections any more.
- cmd/fix resolve_targets: correctly handle when an empty targetlist in -t option is given.
- cmd/chg contrib/edrcupgrade: repeat target input query when an empty target list is given.

- cmd/chg mc, mcview, mcedit: version 4.6.28 compiled for Linux-64/GLIBC 2.27.
- cmd/chg many: cleanup of old, inactive binaries.
- cmd/chg contrib/scriptdocbook: new special reference [----] to add a horizontal line allowing some structuring of entries.
- cmd/chg contrib/scriptdocbook: bibliography fields output order changed to scientific standard.
- cmd/new sortv: sort -V for operating systems where sort lacks this functionality.
- cmd/chg logtail: sort log files by version.
- cmd/chg logtail: beside 'Ctrl+C' abort also when entering '0'.
- cmd/chg logview: sort log files by version.
- cmd/chg logview: beside 'Ctrl+C' abort also when entering '0'.
- cmd/chg contrib/logs: a global search option allows to search in all log files.
- cmd/new xml2json: convert XML file to JSON.
- cmd/new cmdlist: list all available WA2L/edrc commands.
- cmd/chg logtail: allows to select a subset of all logfiles.
- cmd/chg logview: allows to select a subset of all logfiles.
- cmd/chg vsdfml: format text by default as manual page and load it into man. The previous default terminal output is available as TTY output format now.
- cmd/chg exiftool: upgraded to version 12.26.
- cmd/chg vsdfml: avoid big indent of body text in MAN mode.
- cmd/chg csv: convert dos to unix files when using "csv cat".
- cmd/chg is_up: IPv6 support added.
- cmd/chg pmdesc: avoid "cannot stat" errors on non existing include directories.
- cmd/new is_user: check if command is started using the correct user.
- cmd/new rssh: direct or interactive remote ssh invocation.
- cmd/chg edrcinit: no longer use the upstart method on Linux (as seen on Ubuntu) because it is now deprecated.
- cmd/chg edrcinit: write actions and config settings to log file.
- cmd/new contrib/motd: print system wide message of the day.
- cmd/chg input: new TTY option to read/write from/to tty.
- dir/chg edrc/var/dns/djbdns/: remove the djbdns directory by hand because the patchinstall cannot remove it due to the fact that it sits in the protected (STATIC) directory edrc/var/dns/.
- lib/new perlmodules: version 5.16.2 (glibc 2.17) compiled and bundled for Linux-64.
- man/chg ini.1: shell usage examples.
- man/new vadsp.1
- man/new whoisin.1
- man/new whoisin.list.4
- man/new tzdump.3
- man/new job.1
- man/new jobwatch.1
- man/new joblog.1
- man/new logcheckd.interface.3
- man/new dbrep.1
- man/new dbrep.cfg.4
- man/new passwdsyncd_apply.3
- man/new remote_shell.3
- man/new remote_copy.3
- man/chg perlversion.3: command has no option.
- man/new role_option.1
- man/new role_option.cfg.4
- man/new scriptextract.cfg.4
- man/new winexe.1
- man/new passwdcombine.1
- man/new edrcscripts.1m

- man/new hostlist.dat.4
- man/new scriptmenupath.3
- man/chg scripttitle.3: exit status 1 documented.
- man/chg edrcintro.1m: documentation of scriptmenupath.
- man/chg edrcintro.1m: documentation of rcac, rdif, rcomm.
- man/new rosid.3
- man/chg contrib.edrc.1m: contrib.clock.1m added.
- man/chg scriptsequence.1: log to scriptsequence.log.
- man/new csv.3
- man/new mandir.4: directory of online man page resources.
- man/new rcac.1
- man/new rcomm.1
- man/new rdif.1
- man/new loggrep.1
- man/new textblock.3
- man/new ansi2txt.1
- man/chg man/Linux-64/man1/lynx.1: for version lynx version 2.8.9rel.1 (08 Jul 2018).
- man/chg ssh-exec.cfg.4: START_USER description changed.
- man/chg contrib.edrc.1m: documentation to include extra log files in contrib.logviewer.
- man/new awk.1
- man/new awk-bsd.1: BSD version man page included (not linked, not used in HTML, PDF).
- man/new bash.1: GNU 3.2 version included.
- man/new gawk.1: Free Software Foundation version included.
- man/new sh.1: Heirloom Bourne Shell version man page (from <https://sourceforge.net/projects/heirloom/files/heirloom-sh/050706/>) included.
- man/new sh-bsd.1: BSD version man page included (not linked, not used in HTML, PDF).
- man/new ksh.1: RDS Standard version included.
- man/chg compatibility.1: add Linux-64.
- man/new xml2json.3
- man/new cmdlist.1m
- man/new edrcports.4
- man/chg perlenv.3: description of how to write perl scripts that live outside of WA2L/edrc, but profit from all perl modules bundled with WA2L/edrc.
- doc/chg doc/.man/bin/*: cleanup of old unused files.

1.5.54

Sun Nov 1 00:01:00 METDST 2020 release

MAIN IMPROVEMENTS:

- cmd/new scriptsequence: check if scripts are started in sequence and inform or abort if not. This command can be used in an _env file or in a recovery script.
- cmd/chg binprobe: probe (compiled) executables if they are able to start on the current operating system.
- cmd/chg many: compiled for Linux-64/GLIBC 2.8.
- lib/chg expect: elimination of extra start script in lib/<OSID>/.
- man/new many

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.54 (Thu Sept 22 20:42:00 METDST 2020)
- cfg/chg kshrc: cdsw can jump to a subdirectory now.
- cfg/chg os_wrapper/timer.cfg: probe options corrected.
- cfg/chg contrib.edrc.countdown.cfg: use COUNTDOWN_COUNTER_CHARACTER="c" to set the character used to display the countdown.
- cmd/chg thttpd: resolve glibc version.
- cmd/chg msmtpt: compiled for Linux-64/GLIBC 2.8.
- cmd/chg thttpd: compiled for Linux-64/GLIBC 2.8.
- cmd/chg pl: compiled for Linux-64/GLIBC 2.8.

- cmd/chg glibc.version: compiled for Linux/GLIBC 2.8.
- cmd/chg sqlite: compiled for Linux-64/GLIBC 2.8.
- cmd/chg nano: version 2.5.1 compiled for Linux-64/GLIBC 2.8.
- cmd/chg os_wrapper: more controlled abort on 'FATAL: Kernel too old' condition.
- cmd/chg shell: many cd* aliases accept a subdir now.
- cmd/chg shell: the ret command jumps back thru the whole cd spur not only between the two last directories (for this use cd -).
- cmd/new xml2csv: convert XML schema to CSV.
- cmd/chg cfg2html: new version 6.41 for Linux-64.
- cmd/chg cfg2html: new version 6.41 for Linux.
- cmd/chg jq: version 1.5 for Linux.
- cmd/chg os_wrapper: allow multiple switches in EXEC_PROBE_OPTION.
- cmd/chg contrib/edrcupgrade: show patch installation step also in terminal window title.
- cmd/chg contrib/doc: allow to use .SS in O: without loosing the preformatted text output mode.
- cmd/fix contrib/doc: no longer extra lines are printed in HTML output (=ScriptDocBook) when empty lines are present between <pre> ... </pre>.
- cmd/chg contrib/scriptdocbook: css support including TOC and Index buttons as known from the new HTML manual pages added.
- cmd/chg LogViewer: possibility to choose if the Index pane is at top or bottom.
- cmd/chg LogViewer: help page added.
- cmd/chg name: resolve also SVG- and empty files.
- cmd/chg revision: resolve also SVG- and empty files.
- cmd/chg outex: correct logfile resolution when called interactively without specifying a logfile as option.
- cmd/new scriptsequence: check if scripts are started in sequence and inform or abort if not. This command can be used in an _env file or in a recovery script.
- cmd/chg contrib/logviewer: description not is not optional any more.
- cmd/chg edrc: script tempfile also removed when _env file is actively exited using the exit command.
- cmd/chg ssh-exec: internal code improvements.
- cmd/chg binprobe: probe (compiled) executables if they are able to start on the current operating system.
- cmd/fix bget: syntax error on un-escaped braces.
- cmd/new sshpass: a utility designed for running ssh using the mode referred to as "keyboard-interactive" password authentication, but in non-interactive mode.
- cmd/chg contrib/scriptdocbook: URL in *.bib file.
- cmd/fix lbanner: argument parsing improved.
- cmd/fix mail_file: more reliable mail sending configuration when using msmtplib as MAIL_CLIENT.
- cmd/fix contrib/scriptdocbook: mail sending.
- cmd/chg contrib/scriptdocbook: it is now possible to protect the ScriptDocBook archive with a password.
- cmd/chg contrib/scriptdocbook, contrib/logviewer, contrib/scriptpack: more error tolerant email address input.
- cmd/chg contrib/scriptdocbook, contrib/scriptpack: can now also produce password protected archives.
- cmd/chg resolve_targetlist: typing error in usage corrected.
- cmd/new contrib/scriptsequence: (remote)-control the behaviour of the scriptsequence program.
- lib/chg Linux/libs: libraries moved from libs/ to 0.0/ and symlinked.
- lib/chg Linux-64/libs: libraries moved from libs/ to 0.0/ and symlinked.
- lib/chg expect: elimination of extra start script in lib/<OSID>/.
- man/chg os_wrapper.3: added libs/ directory for a specific glibc version.
- man/chg manpages.4: description on how to build the documentation using make.
- man/new scripttitle.3
- man/new scriptsequence.3
- man/new sshpass.1
- man/new portscan.3
- man/new is_writeable.3
- man/new is_existing.3

- man/new filewatch.3
- man/new scriptrevision.3
- man/new outex.1
- man/new perlversion.3
- man/new lbanner.3
- man/new locations.1
- man/new locations.cfg.4
- man/new xtee.1
- man/new xpid.1
- man/new xmore.1
- man/new ssh-keyadd.cfg.4
- man/new ssh-keyadd.pub.4
- man/new streamcat.3
- man/new ksh_wrapper.cfg.4
- man/new perl_wrapper.cfg.4
- man/new xlog.1
- man/new xml2csv.1
- man/chg manpages.4: review.
- man/new is_permitted.3
- man/new manvi.1
- man/new manvi.cfg.4
- man/new repeat.3
- man/new rel2abs.3
- man/new sortc.3
- man/new sectioncat.3
- man/new filelink.1
- man/new hwinventory.1
- man/new is_running.3
- man/new java_wrapper.cfg.4
- man/new rl.1
- man/new log_files.cfg.4
- man/new fields2swvi.3
- man/new usage.list.4
- man/new epub2pdf.1
- man/new passwdsort.1
- man/new resolve_targetlist.3
- man/new pdfscissors.1
- man/new pdfmetaedit.1
- man/new ypxfr_all.1
- man/new java.1
- man/new sw_report.1
- man/new java_wrapper.1
- man/new kshrc.4
- man/new exrc.4
- man/new readline.3
- man/new rsat.1
- man/new rsat.cfg.4
- man/new shell.cfg.4
- man/new sysdiff.1
- man/new pkg_hostname.cfg.4
- man/new edrcinit.handler.3
- man/new sw_inventory.4
- man/new untrash.1

- man/new ini.1
- man/new ini.cfg.4
- man/new env.4
- man/new timezone.dat.4

1.5.53

Sun Sep 20 07:47:12 METDST 2020 release

MAIN IMPROVEMENTS:

- cmd/chg os_wrapper: can handle different glibc versions.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.53 (Mon August 24 11:08:00 METDST 2020)
- cfg/chg edrcman.map: mappings of shell added.
- cmd/chg exiftool: upgraded to version 12.00.
- cmd/chg shell: view alias changed from view to vi -R.
- cmd/chg edrc: probes on *Linux* if HISTORY_BROWSE is supported and only in this case it is enabled.
- cmd/chg os_wrapper: can handle different glibc versions.
- dir/chg edrc/lib/Linux-64: now supports binaries for different glibc versions.
- dir/chg edrc/bin/Linux-64: now supports binaries for different glibc versions.
- dir/chg edrc/src: cleanup of code fragments.
- lib/new perlmodules: version 5.26.1 (glibc 2.27) compiled and bundled for Linux.
- lib/new perlmodules: version 5.10.0 (glibc 2.11.2) compiled and bundled for Linux.
- lib/chg perlmodules: beginning with version 5.26.1 the install directory structure in edrc/lib/perl/pm/perl5/<version> and the build process to bundle perl modules has been significantly simplified. As long as the .perl_wrapper is used to start perl programs, no change in the perl program is needed.
- man/chg os_wrapper.1: description of the GLIBC version dependent binary location.

1.5.52

Sun Jul 26 18:46:40 METDST 2020 release

MAIN IMPROVEMENTS:

- cmd/chg edrc: new environment variable \$EDRC_TMP_DIR_SESSION that points to a secure temporary directory to be used to save temporary data that is used multiple times throughout an edrc session.
- cmd/chg contrib/pwsafe: remember the entered master password between queries.
- cmd/chg csv2worksheet: now can also load data into xlsx files.
- cmd/new ssh-exec: execute commands provided thru stdin on remote system using ssh.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.52 (Sat May 9 14:29:34 METDST 2020)
- cmd/chg contrib/logs: use vi -R instead of view (which sometimes is not a read only vi) to view files.
- cmd/chg contrib/logs: also output and query MENUPOINT.
- cmd/chg contrib/pwsafe: remember the entered master password between queries.
- cmd/chg contrib/pwsafe: allow pwsafe to be called to query password for user@system directly in recovery script (=direct query).
- cmd/chg edrc: use vi -R instead of view (which sometimes is not a read only vi) to view files.
- cmd/chg input: on LOG_STARS also log stars for default input.
- cmd/chg edrc: new environment variable \$EDRC_TMP_DIR_SESSION that points to a secure temporary directory to be used to save temporary data that is used multiple times throughout an edrc session.
- cmd/chg edrc: log also menu point to log files.
- cmd/chg edrcenv: print also \$EDRC_TMP_DIR_SESSION.
- cmd/chg WA2LDocBrowser: let the menu width resize.
- cmd/new ssh-exec: execute commands provided thru stdin on remote system using ssh.
- cmd/chg ssh-keyadd: set correct file permissions if authorized_keys file does not exist in a users home.
- cmd/chg csv2worksheet: now can also load data into xlsx files.
- cmd/new scripttitle: print recovery script title (menupoint and description). This command is intended to be called from a recovery script or _env file to give the user more awareness which menu point

- he started.
- cmd/new shell: cda, cdb, cdc, cdd, cdp to cd to the directory saved in the related working variables \$a, \$b, \$c, \$d, \$p.
- cmd/chg install: print correct path when adjusting permissions.
- cmd/chg contrib/doc: improvement on the remembering of the last doc invocation.
- cmd/chg ccrypt: version 1.11 provided as statically linked executable to avoid missing libraries errors.
- cmd/new contrib/logviewer: create a browseable LogViewer HTML archive containing all log files between two selected dates.
- cmd/chg contrib/scriptdocbook: correct handling of sender address when sending the archive by email.
- cmd/new outex: produce # O: prepended output of a given logfile to be used within recovery scripts in contrib.doc.
- cmd/chg edrcman: suppress the multiple manpage suggestion dialog as observed on SuSE 11.1 Linux.
- cmd/new contrib/tail: to tail -f a file from within edrc.
- cmd/chg edrc: after editing a file no longer provide [n] as default choice to "not to save" the file, now an explicit input is needed.
- cmd/chg choice: lesser number of space between prompt and colon when no default key is given.
- cmd/chg edrc: allow using the cursor keys in edrc prompt to access the history.
- cmd/chg contrib/doc: new T: tag to document the DURATION of a script run.
- cmd/chg edrcman: mapfile to map certain manual pages.
- cmd/new jq: a JSON processor.
- cmd/chg contrib/pwsafe: now also supports semicolons, spaces and tabs in passwords.
- man/chg edrc.1m: document the usage of the \$EDRC_TMP_DIR_SESSION variable.
- man/chg contrib.edrc.1m: document direct query functionality of pwsafe.
- man/chg edrcintro.1: document ssh-exec.
- man/chg EDRC.1: add ssh-exec, ssh-exec.cfg.
- man/new ssh-exec
- man/chg wintoolsintro.1: short description of outex.
- man/chg wintoolsintro.1: short description of jq.
- man/chg contrib.doc.1m: hint to outex usage.
- man/chg contrib.doc.1m: documentation of doc tag T: (DURATION).
- man/new js.3: JSON processor.
- doc/chg doc: directory structure in doc/ cleaned up. If you linked manual pages as for example in *.bib files, as: PATH=doc/edrc_manpages-1.5.51/edrc.html this must be changed to PATH=doc/manuals/man1m/edrc.1m.html.

1.5.51

Tue May 5 07:59:42 METDST 2020 release

MAIN IMPROVEMENTS:

- cmd/chg contrib/*: simplify implementations by the use of \$EDRC_TMP_DIR as temporary directory instead of handling it within the contributed command.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.51 (Tue Mar 17 00:36:28 CET 2015)
- cfg/chg kshrc: add cdtmp alias.
- cfg/chg lotsctl: use \$EDRC_TMP_DIR as temporary directory in all scripts where applicable in entire script tree.
- cfg/chg sys: use \$EDRC_TMP_DIR as temporary directory in all scripts where applicable in entire script tree.
- cfg/chg shell.cfg: new setting USER_DROPBOX_DIR.
- cfg/chg lib/os_wrapper/wget.cfg: EXEC_PRIORITY=OS.
- man/chg shell.1: description of cdtmp command.
- man/chg edrcintro.1: hint to other WA2L projects in NOTES section.
- man/fix edrcsetup.1m: created package in step 6 from edrc_WA21 to edrc_ACME.
- man/chg vsdfml.3: use #!/usr/bin/env vsdfml instead of #!/bin/sh vsdfml as magic key.
- man/chg shell.1: add description of pid command.

- man/cor syspoll.cfg.4: description of LOCK_TIMEOUT.
- man/chg logview.1: only files that can be read by the user are listed.
- man/chg logtail.1: only files that can be read by the user are listed.
- man/chg lsof.1: version 4.88 for Linux.
- man/cor syspoll.1m: correct date format from %Y%d%m to %Y%m%d in examples.
- man/chg syspoll.1m: description that -t option now also accepts hostgroups directly.
- man/chg shell.1: document edrcdebug to compute also a lists of commands.
- man/chg edrcinit.1m: documentation of the enable and disable options.
- man/cor edrcintro.1: short description of cmswitch command.
- man/ch edrcintro.1: short description of untxz command.
- man/new untxz.1
- cmd/fix ux2dos: no error message when '-' is given as input file.
- cmd/fix dos2ux: no error message when '-' is given as input file.
- cmd/chg name: process also release-numbered files as present in edrc/lib/daemon/.
- cmd/chg name: now process the first 20 instead of 10 lines for header information.
- cmd/chg filewatch: avoid error message when file to watch does not exist (yet).
- cmd/chg filewatch: hide cursor and no longer print command.
- cmd/chg filewatch: allow also a process been referenced by process id.
- cmd/chg edrcenv: add output of \$EDRC_TMP_DIR, \$EDRC_CONTRIB_VARDIR.
- cmd/new cdtmp: cd to \$EDRC_TMP_DIR or /tmp if the \$EDRC_TMP_DIR is not set.
- cmd/chg edrc: assign tempdirs created for scripts, contrib commands and shell to edrc to list it active when calling listtemp.
- cmd/chg contrib/dirwatch: use \$EDRC_TMP_DIR as temporary directory, now.
- cmd/chg contrib/logs: use \$EDRC_TMP_DIR as temporary directory, now.
- cmd/chg contrib/scriptdocbook: use \$EDRC_TMP_DIR as temporary directory, now.
- cmd/chg contrib/scriptpack: use \$EDRC_TMP_DIR as temporary directory, now.
- cmd/chg contrib/clonemenu: use \$EDRC_TMP_DIR as temporary directory, now.
- cmd/chg contrib/doc: use \$EDRC_TMP_DIR as temporary directory, now.
- cmd/chg contrib/edrcupgrade: new default URL for download is <https://master.dl.sourceforge.net/project/wa2l-edrc/edrc-patch>.
- cmd/new cdbx: change to user's DropBox directory.
- cmd/new cdrel: change to release control folder (for edrc developer).
- cmd/chg logtail: when calling as non-root users only list the files that the user is permitted to read.
- cmd/chg logview: when calling as non-root users only list the files that the user is permitted to read.
- cmd/chg lsof: version 4.88 for Linux.
- cmd/chg h2: version 1.4.187.
- cmd/chg edrcupgrade: read environment file to set http_proxy, no longer switch to root user.
- cmd/chg hwinventory: improvement in resolving the OSDISTRIBUTION field.
- cmd/chg syspoll: also accept hostgroups (@HOSTGROUP) as target specification in the -t option.
- cmd/chg tsize: use stty internally also on Linux due to SIGTSTP signals received on Ubuntu calling /usr/bin/resize.
- cmd/chg edrcdebug: now more then one command can be specified with one call to edrcdebug supplying a space separated list.
- cmd/chg manvi: exclude .ps and .html files from selection list.
- cmd/chg edrcinit: disable option to disable services startup thru edrc, enable to enable disabled startup again.
- cmd/chg contrib/countdown: allow countdown calculation based on 'Planned End Time'.
- cmd/chg contrib/countdown: more precise output prioritisation on terminal height reduction.
- cmd/chg contrib/countdown: internal design improvements.
- cmd/chg manvi: menu point to quit the command.
- cmd/chg osid: return a 'binary incompatible' error message to stderr if the test run of <osid>.probe is not successful.
- cmd/chg edrcenv: output of environment variables \$OSID_PROBE, \$PRINT_FIT2WIDTH, \$SYSPOLL_CONFIGFILE, \$LOTS_CONFIGFILE, \$LOTS_IDENTITYIES,

\$SYSCONFIG_ROOTDIR, \$SYSCONFIG_FORCE when set.

- cmd/new untxz: uncompressed a *.tar.xz file.
- cmd/chg lscomp, llcomp, catcomp: process *.tar.xz files.
- cmd/chg collectd: removed from package.
- cmd/chg lscomp, llcomp, catcomp: process *.epub files.

1.5.50

Sun Mar 15 19:30:33 CET 2015 release

MAIN IMPROVEMENTS:

- cmd/chg edrc: provide a temporary directory \$EDRC_TMP_DIR during execution of recovery script, contributed command and the shell command.
- cmd/new ux2dos: convert ASCII fileformat between UNIX and DOS formats.
- cmd/new dos2ux: convert ASCII fileformat between UNIX and DOS formats.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.50 (Tue Feb 26 21:54:09 CET 2015)
- cfg/chg kshrc: set ls colors for .css, .js, .awk, Makefile, Makefile.var and COPYING.
- cfg/chg shell.cfg: add LC_TIME=C.
- man/chg edrcintro: add description of edrc/var/manvi/.
- man/chg shell.1: add descriptions of files in FILE section.
- man/chg edrcintro.1: add kshell description.
- man/chg EDRC.1: add kshell.
- man/chg compatibility.1: add kshell.
- man/chg syspoll.1: add csv2worksheet (3) to SEE ALSO.
- man/new dos2ux.3
- man/new ux2dos.3
- man/chg: edrc.1m: description of the \$EDRC_TMP_DIR environment variable.
- cmd/new kshell: print path of Korn- or Bourne Again Shell.
- cmd/chg scripthheadersync: be more error tolerant on files containing spaces.
- cmd/chg lsmv: quote all file names to support also file names with spaces.
- cmd/chg lscp: quote all file names to support also file names with spaces.
- cmd/new ux2dos: convert ASCII fileformat between UNIX and DOS formats.
- cmd/new dos2ux: convert ASCII fileformat between UNIX and DOS formats.
- cmd/chg mail_file: improve portability thru the use of ux2dos instead of native command.
- cmd/chg contrib/edrcupgrade: allow cancel in select patchfile to install dialog.
- cmd/chg edrc: provide a temporary directory \$EDRC_TMP_DIR during execution of recovery script, contributed command and the shell command.
- cmd/chg manvi: provide man page templates.
- cmd/cor syspoll: correct usage output of '-s set' option.

1.5.49

Mon Feb 23 23:16:08 CET 2015 release

MAIN IMPROVEMENTS:

- cmd/new syspoll: poll systems for actions and execute queries.
- opt/new lots: -a list_session to list all lots sessions.
- cmd/new wmic: client to use the Windows Management Instrumentation from Linux.
- cmd/new winexe: client to execute commands remote on Windows.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.49 (Sun Nov 16 23:47:04 CET 2014)
- cfg/new manvi.cfg: configuration file for manvi.
- man/chg edrcrevision.cfg.4: add apprevision(3) to SEE ALSO.
- man/chg edrcintro.1: manvi description added.
- man/chg EDRC.1: manvi(1), manvi.cfg(4) added.
- man/chg indent.3: allow output channel setting of leading indent.
- man/new syspoll.1
- man/chg edrcintro.1: sectioncat description added.

- man/chg edrcintro.1: syspoll description added.
- man/chg compatibility.1: syspoll and sectioncat added.
- man/chg sav.1: add */lib/syspoll/protocols/* to special sav handling.
- man/chg daemon_wrapper.1: add documentation of exit code 103 (permission denied to start command).
- man/chg os_wrapper.1: add documentation of exit code 103 (permission denied to start command).
- man/chg perl_wrapper.1: add documentation of exit code 103 (permission denied to start command).
- man/chg ksh_wrapper.1: add documentation of exit code 103 (permission denied to start command).
- man/new undeb.1
- man/new syspoll.cfg.4
- cmd/new manvi: edit WA2L/edrc manual pages.
- cmd/chg edrcman: error output on missing man page.
- cmd/chg indent: allow output channel setting of leading indent.
- cmd/chg xpid: option parsing changed. If no dash option is given select query string from command list, else parse options as given.
- cmd/new sectioncat: cat a section ([SECTION]) of a (config) file.
- cmd/new syspoll: poll systems for actions and execute queries.
- cmd/fix ssh-keyadd: removed unwanted key output that included an egrep error message when adding keys.
- cmd/chg sav: */lib/syspoll/protocols/* added for special sav handling.
- cmd/chg .daemon_wrapper: check if calling user has the permissions to execute/access started file.
- cmd/chg .ksh_wrapper: check if calling user has the permissions to execute/access started file.
- cmd/chg .os_wrapper: check if calling user has the permissions to execute/access started file.
- cmd/chg .perl_wrapper: check if calling user has the permissions to execute/access started file.
- cmd/chg edrcman: output when there is no manpage for a WA2L/edrc command available.
- cmd/new wmic: client to use the Windows Management Instrumentation from Linux.
- cmd/new winexe: client to execute commands remote on Windows. from Linux.
- cmd/new undeb: uncompress deb (Debian package) files.
- cmd/cor watchdog: typing error in usage message corrected.
- opt/new lots: -a list_session to list all lots sessions.
- lib/new Text::Aspell: perl module for spell checking using aspell.
- lib/new Net::SNMP, Crypt::DES, Crypt::Rijndael, Digest::SHA1: modules for SNMP queries.

1.5.48

Sun Nov 16 20:27:45 CET 2014 release

MAIN IMPROVEMENTS:

- man/chg many manpages: documentation enhancements.
- cmd/chg sqlite, h2: version upgrades.
- lib/chg libpng, libz: version upgrades.
- lib/new Tk: integration of Tk::* perl modules for graphical programming in perl.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.48 (Sun Oct 26 02:05:03 CEST 2014)
- cfg/new nginx.doc.cfg: provide nginx configuration to serve var/www/doc.
- man/new scriptgrep.1
- man/new filegrep.1
- man/new histlist.1
- man/chg EDRC.1: add scriptgrep (1), filegrep (1), histlist (3).
- man/chg compatibility.1: edrcinit specification.
- man/chg compatibility.1: table format modernized.
- man/chg edrcman.1: add description when a missing *(PK man page is requested).
- man/new edrcinit.1m
- man/chg manpages.4: elimination of dq (double quote) and aq (single quote) macros to improve output of generated html files.
- man/chg histlist.3 fssum.cfg.4 title.1 fmatch.3 WA2L::Util.3 manpages.4 edrcrevision.1 edrcintro.1 perlenv.3 perl_modules.3: avoid the use of the ad and dq man macros to improve output of generated html files.

- man/chg edrc.1m: var/tmp directory description added.
- man/new edrcinit.cfg.4
- man/chg sav.1: add /etc/default/ for special save file handling.
- man/chg EDRC.1: title page layout changed.
- cmd/chg edrcinit: change ownership of rc script on selected operating systems.
- cmd/chg edrcinit: support rcinstall on upstart Linux systems, rcinstall simplified.
- cmd/chg edrcman: a suggestion to invoke 'edrcman manpage' is printed, when a currently non existent WA2L/edrc man page is requested.
- cmd/chg sav: /etc/default/ added for special file save handling.
- cmd/chg h2: new version 1.4.182 (2014-10-17).
- cmd/new filewatch: watch the progress of file size.
- cmd/chg sqlite: new SQLite version 3.8.7.1.
- cmd/chg WA2LDocBrowser: menu and foot line formatting changes.
- dir/new var/tmp/: to store temp files in a more controlled environment.
- lib/chg lib/perl/pm: eliminate world writable files and directories.
- lib/chg libpng: new version 1.6.14.
- lib/chg libz: new version 1.2.8.
- lib/new Tk: integration of Tk::* perl modules for graphical programming in perl.

1.5.47

Sat Oct 25 23:16:48 CEST 2014 release

MAIN IMPROVEMENTS:

- man/chg many manpages: documentation enhancements.
- cmd/chg many commands: improved portability.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.47 (Wed Mar 26 22:50:27 CET 2014)
- cfg/chg foswiki/.../LocalSite.cfg: also handle DefaultUriHost in edrcinit/WA2Ledrc.foswiki handler.
- man/chg edrcsetup.1m: formulation in NOTES section enhanced.
- man/chg busy.1: add description of \$BUSY_ANIMATION.
- man/cor os_wrapper.1: path in example corrected.
- man/cor edrcsetup.1m: comment in example hostlist.dat file output corrected from hostlistdat2cfg -a dat to (...) -a list_dat.
- man/cor ssh-keyadd.1m: SEE ALSO ssh-keygen.
- man/chg perl_modules.3: reduce perl modules list to modules used by distributed commands written in perl.
- man/chg edrcintro.1: add lotsctl as example configurations in sat(1) explanation.
- man/chg edrc.1m: documentation of 'usage' command as alias to 'help'.
- man/chg pf_wrapper.1m: description of sudoers example enhanced.
- man/cor edrcsetup.1m: sudoers example configuration file corrected.
- man/chg shell.1: description of manvi added.
- man/new eshell.1
- man/chg perl_modules.3: documentation of how to compile and install bundled Perl modules if needed.
- man/chg compatibility.1: compatibility table format changed.
- man/chg edrcsetup.1m: add Ubuntu 14.04 as additional development operating system.
- man/chg perl_modules.3: change pack example to use the standard software location edrc/var/sw for package output.
- cmd/chg contrib/countdown: print halted and stopped state in banner to allow status view from distance.
- cmd/chg busy: can now print also different animations then the classic rotating slash by setting the BUSY_ANIMATION environment variable.
- cmd/new lbanner: print a large ASCII banner.
- cmd/chg leo: output formatting.
- cmd/chg contrib/leo: input method.
- cmd/new pdfmetaedit: GUI to edit PDF file meta data.
- cmd/new pdfscissors: GUI to crop PDF file for eBook reader.

- cmd/chg edrc: command 'usage' as alias to 'help'.
- cmd/fix apply2sw_inventory calc choice cltrash cmmon countdown crfile doc drvstat edrcman envpasswd-strip filedist filelink filesize freespace fssum histlist hwinventory input is_config_byhand job listtemp locations logcheckd logs lots mail_file maketemp makeuser omnimon omnistat omniutil passwdsyncd passwdsyncd_apply pscount pslist rcmd remote_copy remote_shell removetemp resolve_targetlist role_option rsat scriptheadersync scriptstats server_environment sessions ssh-keyadd swvi trash untrash user_info watchdog whereami: set absolute path to echo binary for Linux.
- cmd/fix pack: correct un-terminated quoted string used in input command options.
- cmd/chg symlink: error handling if change directory to directory where the symlink should be created is not possible.
- cmd/chg edrcinit: also support rc*.d directories as located on Ubuntu Linux systems.
- cmd/new lsof: lsof_4.87_3.13.0-37-generic for Ubuntu.
- cmd/chg lscomp: debian package (*.deb) support.
- cmd/chg llcomp: debian package (*.deb) support.
- cmd/chg catcomp: debian package (*.deb) support.
- lib/new Text::vCard: a package to edit and create a single vCard (RFC 2426).

1.5.46

Tue Mar 25 01:53:42 CET 2014 release

MAIN IMPROVEMENTS:

- cmd/new shellinaboxd: publish command line shell through AJAX interface.
- cmd/new leo: command line interface to dict.leo.org.
- doc/new toc: Table of Contents with page numbers in PDF document.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.46 (Fri Mar 7 08:36:34 CET 2014)
- cfg/chg kshrc: change vi/vim startup to also allow editing of files having spaces.
- man/cor symlink.1: the link is removed, not the original.
- man/new leo.1
- man/chg contrib.edrc.1m: description of contrib.leo.1m added.
- man/chg pid.1: add pslist(3) and xpid(1) to see also.
- man/cor vsdfml.3: troff formatting to view section MARKUP TAGS at the right position.
- man/new shellinaboxd.3: publish command line shell through AJAX interface.
- man/chg edrcsetup.1m: give hint to edrcupgrade command for patch download and install.
- man/chg edrcsetup.1m: add description of pbin directory to ease up startup of commands thru sudo or pfexec.
- man/cor busy.1: description grammar.
- doc/new toc: Table of Contents with page numbers in PDF document.
- cmd/chg hostlistdat2cfg: accept the XGRPS_PREF=str option in -m, as documented in the man page and in the commands usage message. The actually implemented option XGRPS_PREFIX=str is also still accepted (and will not be removed in the future) for backward compatibility to ensure users using that option can continue to use the command without change.
- cmd/fix symlink: setting and querying of symlink ownership when not called by root.
- cmd/new leo: command line interface to dict.leo.org.
- cmd/new contrib/leo: access to leo in edrc.
- cmd/chg edrc: check prompt input for allowed characters.
- cmd/new shellinaboxd: publish command line shell through AJAX interface.
- cmd/new xpid: convenience function to pslist and enhanced version to the pid command.
- cmd/chg contrib/countdown: avoid flicker on output.
- cmd/chg lib/edrcinit/WA2L.*: add EDRC_DEBUG_* facility, from now on the edrcdebug command can be used to debug those service handlers.
- cmd/fix days: parsing of date format YYYY-MM-DD.
- cmd/fix tsize: check if resize command does not exist.

1.5.45

Thu Mar 6 23:23:01 CET 2014 release

MAIN IMPROVEMENTS:

- cmd/chg symlink: setting of symlink ownership based on replaced link or original file.
- cmd/new exiftool: to edit/view meta information in files.
- cmd/new epub2pdf: convert EPUB files to PDF.
- cmd/chg contrib/countdown: visual and functional improvements.
- cmd/chg h2: integration of version 1.3.175.
- cmd/chg cfg2html: integration of version 5.31 for HP-11, HP-11i, HP-11ia.
- man/chg *: documentation enhancements.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.45 (Sat Aug 24 12:00:49 CEST 2013)
- cfg/chg usage.list: nmap, nping, h2 added.
- cfg/chg edrcinit.cfg: entry for sshd_system added.
- cfg/chg kshrc: .ico files added.
- cfg/fix kshrc: save working variables \$a, \$b, \$c, \$d, \$p with quotes to enable settings containing spaces to be loaded correctly.
- man/new csvcat.1
- man/chg pf_wrapper.1: description of exit state 107.
- man/chg contrib.edrc.1m: the scriptpack.fl file should be located in the \$EDRC_SCRIPTS_BASEDIR and not in \$EDRC_ENTRY_DIR for contrib.scriptdocbook(1m).
- man/chg tty_columns.3: tty_variable is used internally.
- man/chg contrib.edrc.1m: configuration of countdown in the optional config file etc/contrib.edrc.countdown.cfg.
- man/chg edrcintro.1: add epub2pdf short description.
- man/chg edrcintro.1: add eshell short description.
- man/chg edrcintro.1: add tput_examples short description.
- man/chg edrcintro.1: remove all perl modules from TOC.
- man/chg EDRC.1: add epub2pdf.
- man/chg EDRC.1: add eshell.
- man/chg EDRC.1: remove all perl modules from TOC, link in HTML output will go via perl_modules.1 man page.
- man/chg compatibility.1: add epub2pdf.
- man/chg compatibility.1: add eshell.
- man/cor perl_modules.3: formatting.
- man/new tput_examples.3: examples of tput(1) usage.
- man/chg textcolor.3: add tput_examples(3) to SEE ALSO.
- man/chg symlink.1: document also setting of owner:group setting of created symlinks.
- man/chg contrib.edrc.1m: add short description of contrib.edrcenv.1m.
- man/chg edrcsetup.1: remove the hostlist.cfg "inline" example of configuring hostgroups and lists within the hostlist.cfg.
- man/new exiftool.1
- doc/chg doc/.man/sourceforge.net/project-web/WA2Ledrc.html: hyperlink cleanup and new DOWNLOADS chapter added.
- doc/new h2-1.3.175.html: HTML and PDF documentation of H" database engine.
- doc/chg edrc_manpages-*.pdf: add date to title page.
- cmd/chg .pf_wrapper: avoid direct call of .pf_wrapper.
- cmd/chg contrib/scripttree: handle symlinks that are broken.
- cmd/chg contrib/scriptpack: ensure proper directory permissions of copied files when packing config files and files listed in scriptpack.fl.
- cmd/chg contrib/scriptdocbook: read scriptpack.fl when located in \$EDRC_SCRIPTS_BASEDIR instead of \$EDRC_ENTRY_DIR.
- cmd/chg contrib/scriptdocbook: do not follow symlinked menus.

- cmd/chg contrib/checklist: do not follow symlinked menus.
- cmd/new lib/edrcinit/WA2Ledrc.sshd: service handler for sshd to be used on systems where sshd is not automatically started.
- cmd/chg contrib/edrcupgrade: allow cancel also when specifying a patchfile to install.
- cmd/cor contrib/countdown: typing error in prompt.
- cmd/chg contrib/countdown: output of help/usage message.
- cmd/chg contrib/countdown: countdown output.
- cmd/chg contrib/countdown: countdown handling streamlined.
- cmd/chg contrib/countdown: output of large countdowns also in large letters displayed correctly.
- cmd/chg contrib/countdown: reads color definitions from optional config file etc/contrib.edrc.countdown.cfg.
- cmd/fix edrcman: always write cache content in language setting 'C' to avoid re-evaluation of cache when the system language differs from 'C'.
- cmd/new epub2pdf: convert EPUB files to PDF.
- cmd/chg h2: new version 1.3.175.
- cmd/new pbin/eshell: normal start of bin/shell.
- cmd/fix a, b, c, d, abc, abcd: save working variables with quotes to enable settings containing spaces to be loaded correctly.
- cmd/fix edrc: \$TODAY is now also exported to the contributed commands.
- cmd/chg symlink: symlink also sets the owner and group permissions of the created symlink whenever possible.
- cmd/new contrib/edrcenv: display official WA2L/edrc environment variables, as the edrcenv command in the shell.
- cmd/chg HP-11ia/cfg2html: new version 5.31.
- cmd/chg HP-11i/cfg2html: new version 5.31.
- cmd/chg HP-11/cfg2html: new version 5.31.
- cmd/new exiftool: view/edit file meta information.

1.5.44

Sat Aug 24 00:11:06 CEST 2013 release

MAIN IMPROVEMENTS:

- cmd/chg www/foswiki: integration of Foswiki-1.1.8 with more flexible file structure and enhanced startup handling thru edrcinit.
- dir/new edrc/pbin: start commands thru RBAC or sudo transparently. User only has to add ~edrc/pbin to \$PATH.
- cmd/new h2: integration of H2 database engine.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.44 (Wed Apr 17 21:54:49 CEST 2013)
- cfg/chg www/report/.meta/VERSION: 1.0.19.
- cfg/chg kshrc: color *.bib and *.fl as config file.
- cfg/chg kshrc: remove mv2dsav alias.
- cfg/chg edrcman.cfg: LC_ALL to ensure proper special character display. If not set, but \$LANG on system is set, LC_ALL is set to \$LANG. If \$LANG also is not defined, the "C" locale is set as default.
- cfg/new h2.cfg: configuration file for H2 database engine.
- cfg/chg edrc.cfg: EDRC_HOSTPROMPT to display the host name after the @ sign in the prompt in full length (long) or to display only the host name part (short).
- man/chg EDRC.1: no longer include the PostScript manpage file.
- man/chg contrib.edrc.1m: add description of checklist to section of contrib.scriptdocbook(1m).
- man/chg WA2Ledrc:report.1: documentation of initial password of the user "admin" in the portal.
- man/chg Linux/banner.3: Copyright to 2013.
- man/new hostlistdat2cfg.3
- man/chg edrcman.1: add description of \$LANG and \$LC_ALL environment variable.
- man/cor *: PDF file of all man pages no longer contains empty pages.

- man/chg name.1: add description of etc/name.index file.
- man/chg edrcrevision.1: add bundled perl modules to lib/* section.
- man/chg shell.1: add cdwww.
- man/new h2.1
- man/new pf_wrapper.1
- man/chg shell.1: description of cdpbin command.
- man/new nping.1
- man/new nmap.1
- doc/chg edrc_manpages<version>.pdf: add graphical titlepage.
- doc/chg edrc_manpages<version>.ps: no longer include the PostScript file into the release. PostScript is still used to create the pdf file.
- cmd/chg www/report/main/help.html: correct logos.my to mylogos and change formatting of file paths.
- doc/new h2-1.3.172.html: HTML and PDF documentation of H" database engine.
- cmd/chg sys/1:patchinstall: enable startup of edrcupgrade contrib command from within that menu point.
- cmd/fix contrib/edrcupgrade: also accept 'c' to cancel in step 'list contents of patchfile'.
- cmd/chg DocBrowser: footer line.
- cmd/chg www/report/cgi/statistics: do not count number of lines in hour statistics.
- cmd/fix lotsctl/e:edit_lotsctl_settings: edit lotsctl.cfg file that as defined in the topmost _env file.
- cmd/chg contrib/scriptdocbook: also add checklist as produced by the contrib.checklist(1m) command to the ScriptDocBook. Permission related to this functionality is contrib.scriptdocbook.checklist.
- cmd/chg contrib/scriptdocbook: read also settings from optional config file
\$EDRC_SCRIPTS_BASEDIR/scriptdocbook.cfg to specify some defaults script tree specific.
- cmd/chg contrib/scriptdocbook: no default on 'create ScriptDocBook' dialog to avoid runaway cancel.
- cmd/chg contrib/scriptdocbook: formatting and style of index page.
- cmd/chg contrib/checklist: all informational output goes to stderr now, to allow contrib.scripttree(1m) to process the output.
- cmd/chg httpd: only list applications that also have a corresponding config file edrc/etc/htthpd.<app>.cfg.
- cmd/chg edrcman: set LC_ALL to ensure proper special character display.
- cmd/chg name: when the description of a file cannot be resolved from file header and ..name_index the global etc/name.index file is consulted.
- cmd/new cdwww: alias to go to var/www.
- cmd/chg DocBrowser: add a favicon icon.
- cmd/chg sysconfig: avoid backing up files multiple times. This was the case when a file has been listed multiple times in the backup file.
- cmd/new h2: H2 database engine console.
- cmd/new .pf_wrapper: start WA2L/edrc commands thru sudo or pfexec.
- cmd/new cdpbin: cd to edrc/pbin.
- cmd/chg edrc: dynamic number of columns on output when invoking edrcperm command.
- cmd/chg edrc: host name prompt part is configurable using the EDRC_HOSTPROMPT setting in edrc.cfg.
- cmd/chg www/foswiki: integration of Foswiki-1.1.8 and change of structure.
- dir/new pbin: location of command symlinks started thru .pf_wrapper.
- lib/new version, Module::Module-Metadata, Perl::Perl-OSType, CPAN::CPAN-Meta-Requirements, CPAN::CPAN-Meta-YAML, JSON::JSON-PP, Parse::Parse-CPAN-Meta, CPAN::CPAN-Meta, Locale::Locale-Maketext, Params::Params-Check, Module::Module-Corelist, Module::Module-Load, Module::Module-Load-Conditional, CPAN::CPAN-Meta, IPC::IPC-Cmd, ExtU-tils::ExtUtils-CBuilder, CPAN::CPAN-Meta-YAML, Parse::Parse-CPAN-Meta, Perl::Perl-OSType, Module::Module-Build, Try::Try-Tiny, Test::Test-Fatal, HTML::HTML-Tree

1.5.43

Fri Apr 12 21:41:54 CEST 2013 release

MAIN IMPROVEMENTS:

- cmd/new hostlistdat2cfg: generate the hostlist.cfg structure based on a simple csv list in hostlist.dat. The use of this command in hostlist.cfg reduces the configuration effort of hostlists and hostgroups significantly.

- cmd/new contrib/edrcupgrade: provide download of WA2L/edrc patches from sourceforge.net, deployment and installation in a contributed edrc command.
- cmd/new lotsctl: a "recovery" script tree to control lots (long term data save).
- cmd/new nginx: the small and portable proxy server is bundled with WA2L/edrc now. The start/stop/reload is handled thru edrcinit.
- cmd/new djbdns: the small and portable DNS server and cache is bundled with WA2L/edrc now. The start/stop/reload is handled thru edrcinit.
- cmd/chg syscp: now allows also to copy a system file to the repository if it is not already present in it.
- cmd/chg WA2L/report: several functional and stability enhancements.
- man/chg edrcsetup.1m: complete review of the WA2L/edrc "Quick Installation Guide".
- man/chg *: many documentation enhancements and corrections.

ALL CHANGES:

- cfg/chg edrcrevision.cfg: new release 1.5.43 (Thu Nov 1 22:24:38 CET 2012)
- cfg/new edrc.lotsctl.cfg: configuration for lots control menu (edrc short start lotsctl).
- cfg/new lotsctl.cfg: general configuration for lots control menu (edrc short start lotsctl).
- cfg/chg www/report/.meta/VERSION: 1.0.18.
- cfg/chg bin/usage.list: add lotsctl, screen.
- cfg/chg edrcman.cfg: add legacy linux man page directories.
- cfg/chg kshrc: only print warning when calling "m" when shell is started within "edrc".
- cfg/chg: edrc.cfg: new setting SCRIPTS_SHELLFALLBACKS to define fallback shells for recovery scripts.
- cfg/chg thttpd.report.cfg: no longer check referer urlpat, to be able to serve the page without many host-name restrictions.
- cfg/chg lib/usage.list: add nc, fcreate, tzdump.
- cfg/new lib/os_wrapper/lsof.cfg
- cfg/chg kshrc: cdsw alias performance.
- cfg/fix kshrc: ensure that termsize always sets the terminal size on initial command start.
- cfg/chg thttpd.report.cfg: change logfile from var/log/thttpd_access.log to var/log/thttpd_report_access.log.
- cfg/new tinydns.cfg: config file for tinydns startup via edrcinit.
- cfg/new tinydns.dat: DNS source data file for tinydns startup via edrcinit.
- man/chg EDRC.1: add lotsctl (1).
- man/chg EDRC.1: add WA2Ledrc:report (1).
- man/chg EDRC.1: add thttpd.report.cfg (4).
- man/chg EDRC.1: add textcolor (3), perlversion (3).
- man/chg edrcintro.1: add lotsctl short description.
- man/chg edrcintro.1: add textcolor short description.
- man/chg edrcrevision.1: definition of patchlevel change in RELEASE-NUMBERS section.
- man/chg edrcman.1: description of the -s option.
- man/new WA2Ledrc:report.1: description of the operating system report portal.
- man/chg edrcintro.1: add WA2Ledrc:report.
- man/chg edrcintro.1: add thttpd.report.cfg (4) in SEE ALSO section.
- man/new pscount.3
- man/new pscount.4
- man/new ssh-keyadd.1m
- man/chg regexintro.4: portability hint in POSIX CHARACTER CLASSES section.
- man/chg edrcintro.1: change email addresses @eds.com to @hp.com, remove email address of Reimund Mueller.
- man/chg edrcintro.1: add hint to Sourceforge.
- man/chg lsof.1: version 4.87 for Linux.
- man/chg edrcrevision.1: new object 'lib'.
- man/chg sav.1: add manual page and *.d directories to special handled dirs.
- man/new cmswitch.1m

- man/new cmmn.1m
- man/new title.1
- man/chg edrc.1m: add validation of magic key of executable recovery scripts.
- man/chg edrc.cfg.4: describe the SCRIPTS_SHELLFALLBACKS setting used to define the magic key shell fall-backs.
- man/new fcreate.3
- man/new textcolor.3
- man/chg contrib.edrc.1m: add the description of the possible deny functionalities of the contrib.count-down command.
- man/chg compatibility.1: add textcolor, nginx, csvcat commands.
- man/new tsize.3
- man/new random.3
- man/chg contrib.edrc.1m: contrib.scriptpack section.
- man/fix contrib.edrc.1m: contrib.scriptdocbook section.
- man/chg edrcintro.1: add untbz, untbz2, uncbz, uncbz2, perlversion short descriptions.
- man/chg edrcintro.1: add hostlistdat2cfg short descriptions.
- man/new uncbz.1
- man/new uncbz2.1
- man/new untbz.1
- man/new untbz2.1
- man/chg patchinstall.1m: description of the -m and -l options.
- man/chg uncgz.1: add references to uncbz, uncbz2, untbz, untbz2 to SEE ALSO.
- man/chg untgz.1: add references to uncbz, uncbz2, untbz, untbz2 to SEE ALSO.
- man/chg unrpm.1: add references to uncbz, uncbz2, untbz, untbz2 to SEE ALSO.
- man/chg unczip.1: add references to uncbz, uncbz2, untbz, untbz2 to SEE ALSO.
- man/chg edrcman.1: add description of influence of config file etc/perl_wrapper.cfg.
- man/chg perlenv.3: add description of directory structure in edrc/lib/perl/pm/ and include an extract of the mode how perl accesses modules of different versions from cpan.org.
- man/new nginx.3
- man/chg nginx.3: use more portable man macros.
- man/chg perl_modules.3: add URI perl module.
- man/chg logtail.1: add description of \$APPROOT resolution in logfile definition.
- man/chg logview.1: add description of \$APPROOT resolution in logfile definition.
- man/chg contrib.edrc.1m: add description of the optional config file contrib.edrc.scriptdocbook.cfg.
- man/chg contrib.edrc.1m: add description of the optional config file contrib.edrc.scriptpack.cfg.
- man/chg edrcsetup.1m: complete review.
- man/chg syscp.1: description of the checking of replacement tags in repository files.
- man/chg man/all/man*/*: add section title to header.
- man/chg syscp.1: add description of adding new files to a repository. Enhance EXAMPLES.
- man/chg shell.1: description of lscolors command.
- man/cor fmatch.3: change double quote macro use to increase man page display portability.
- man/new ipcalc.1
- man/chg perl modules: no longer distribute the manpages of bundled perl modules in the edrc/man/all/man3/ directory. The manpages of the bundled perl modules are only located in the edrc/lib/perl/pm/perl5/share/ directory.
- man/chg hostlist.3: add hostlistdat2cfg (3) to SEE ALSO section.
- man/chg hostlist.cfg.4: add hostlistdat2cfg (3), hostlist.dat (4) to SEE ALSO section.
- man/chg name.1: add description of ..name.index file.
- man/new strace.1:
- man/chg duvi.1: add exrc to FILES section.
- man/chg swvi.1: add exrc to FILES section.
- man/chg logview.1: add exrc to FILES section.
- man/chg sysvi.1: add exrc to FILES section.

- man/chg rcmd.1: add exrc to FILES section.
- man/chg mail_file.cfg.4: documentation of the options MAIL_SERVER_PORT, MAIL_SERVER_NAME, MAIL_CLIENT_OPTS, MAIL_CLIENT.
- doc/chg doc/*-current*: no longer create, maintain and distribute the symlinks to the current documents.
The most recent documents can be accessed always thru the DocBrowser.
- cmd/fix .perl_wrapper: also check if the executable found as a shell is a file.
- cmd/fix .ksh_wrapper: also check if the executable found as a shell is a file. cmd/new lotsctl: symlink to sat to provide a shortstart for the lots control menu.
- cmd/fix www/report/cgi/users.modify: avoid that a non admin user can modify the user account.
- cmd/new www/report/cgi/settings: dynamic settings menu which displays menu points based on user role.
- cmd/new www/report/cgi/settings.list: display the portal settings.
- cmd/chg www/report/cgi/reports: display the customers logo in header if it exists.
- cmd/chg www/report/cgi/doc: display the customers logo in header if it exists.
- cmd/chg www/report/cgi/doc: only list files, that have the correct permissions to be served (file=r.., directory=r.x).
- cmd/chg www/report/cgi/cgi.inc: add get_logo to library.
- cmd/chg www/report/cgi/logo: move get_logo to library.
- cmd/chg www/report/cgi/about: copyright date to 2012.
- cmd/chg www/report/cgi/statistics: back-links to dynamic home menu.
- cmd/chg www/report/cgi/users.*: back-links to dynamic user administration menu.
- cmd/chg www/report/cgi/logo.gallery: back-links to dynamic home menu.
- cmd/new www/report/cgi/home: dynamic home menu which displays menu points based on user role.
- cmd/new www/report/cgi/users: dynamic user administration menu which displays menu points based on user role.
- cmd/chg www/report/*: change all hyperlinks from absolute to relative.
- cmd/chg www/report/main/menu.html: provide a running clock.
- cmd/fix www/report/cgi/*: no longer check \$HTTP_REFERER in cgi scripts.
- cmd/fix www/report/cgi/statistics: calculation of user access and selection of http server accesses.
- cmd/chg www/report/cgi/statistics: new statistics of accesses per hour of the day only.
- cmd/chg uniqpath: eliminate repetitive colons (::) in output.
- cmd/fix contrib/doc: internal call of edrcman instead of man to display the documentation menu points to ensure that the documentation display is more reliable.
- cmd/fix ssh-keygen: correct authorized_keys writing in header and usage.
- cmd/chg contrib/countdown: output screen adjusted, implementation of halt and continue functionality to temporarily suspend the countdown.
- cmd/chg contrib/countdown: provides now the functionalities init, start, stop, resume, display to be used in edrcperm.
- cmd/chg HP-11ia/cfg2html: new version 5.27.
- cmd/chg HP-11i/cfg2html: new version 5.27.
- cmd/chg HP-11/cfg2html: new version 5.27.
- cmd/chg Solaris/cfg2html: new version 1.7.0.
- cmd/chg Solaris-x86/cfg2html: new version 1.7.0.
- cmd/chg Linux/lsof: new version 4.87.
- cmd/chg sav: handle files in man directories special.
- cmd/chg sav: more flexible and more general special directory handling.
- cmd/chg cmswitch: logfile when not using within edrc is edrc/var/log/cmswitch.log.
- cmd/chg edrc: copyright 2012 added. Validate if the interpreter for the recovery script exists and only start it if it does.
- cmd/chg edrc: reset windows title after recovery script and contributed command start or abort.
- cmd/chg edrc: if the shell in the magic key of a recovery script does not exist, a fallback mechanism tries to resolve alternate shell locations and alternate shells to ensure a script start. Script that do not have execute permissions are no longer started simply using the ksh shell, here also the fallback mechanism that resolves the magic key and the alternate shells, applies.

- cmd/chg edrc: output of recovery scripts and contributed commands to sdterr are now saved to the log files, too.
- cmd/new textcolor: to set the terminal text color.
- cmd/chg shar.setup: add information to transfer the package in binary mode to header.
- cmd/chg contrib/license: copyright 2003-2013.
- cmd/chg contrib/scriptpack: now it is also possible to select to save all config files and a list of other files to the script package, not only the related edrc.cfg file.
- cmd/new untbz: unzip a b2zipped tar file.
- cmd/new untbz2: unzip a b2zipped tar file.
- cmd/new uncbz: unzip a b2zipped cpio file.
- cmd/new uncbz2: unzip a b2zipped cpio file.
- cmd/chg .perl_wrapper,perlenv: PERLLIB resolution improved. With MODULE_DIR in perl_wrapper.cfg the location of the perl modules can be set (intended for testing purposes).
- cmd/chg edrcman: display also man pages of bundled Perl modules. To achieve this, also the MODULE_DIR setting of perl_wrapper.cfg is read.
- cmd/chg lib/perl/pm: compiled all bundled Perl modules for perl 5.14.3 and the x86_64-linux-thread-multi architecture.
- cmd/new lib/perl/pm: new module URI, Schedule::Cron, Time::DaysInMonth, Time::ParseDate, Time::CTime, Time::Timezone, Time::JulianDay.
- cmd/new nginx: HTTP and reverse proxy server, mail proxy server.
- cmd/chg name: when file header does not provide an entry for a symlinked file, display the header description of the original file. When file is a saved file, also resolve file header.
- cmd/chg name: if the description cannot be resolved from the file header, consult also the ..name.index file for a description.
- cmd/fix edrcinit: usage message corrected.
- cmd/chg edrcinit: rcinstall improved.
- cmd/chg edrcinit: status output includes also the guarded information, now.
- cmd/new lib/edrcinit/WA2Ledrc.nginx: new handler for edrcinit.
- cmd/cor histlist: file header.
- cmd/chg logview: resolve \$APPROOT (=‘approot’) environment variable in logfile definition.
- cmd/chg logtail: resolve \$APPROOT (=‘approot’) environment variable in logfile definition.
- cmd/chg histlist: suppress output of last column having undefined column.
- cmd/chg contrib/scriptdocbook: save the persistent input data now in a single file.
- cmd/chg syscp: check for replacement tags @[-_A-Z]+@ of differing files in the repository and display a warning to inform.
- cmd/chg syscp: can now also add new files to repositories.
- cmd/new lscolors: toggle ls color display in shell between true and false.
- cmd/chg is_running: remove s-bit, due to the fact it never worked really.
- cmd/chg patchinstall: avoid blocks output when printing meta data or list files of patch.
- chg/cmd contrib.logs: performance improvement on cache file handling.
- chg/cmd bin/Linux*/lsof: do a more fine os check.
- chg/cmd .os_wrapper: export \$OSID. This enables to act operating system specific in the lib/os_wrapper/<cmd>.cfg files.
- cmd/new perlversion: print resolved perl interpreter version.
- cmd/chg contrib/cmswitch: add startup prompt.
- cmd/chg contrib/dirwatch: add startup prompt.
- cmd/chg contrib/drvstat: add startup prompt.
- cmd/chg contrib/logtail: add startup prompt.
- cmd/chg contrib/oerr: add startup prompt.
- cmd/chg contrib/omnimon: add startup prompt.
- cmd/chg contrib/omnistat: add startup prompt.
- cmd/chg contrib/pid: add startup prompt.
- cmd/chg contrib/pstree: add startup prompt.

- cmd/chg contrib/scriptdocbook: add startup prompt.
- cmd/chg contrib/scriptpack: add startup prompt.
- cmd/chg contrib/scripttree: add startup prompt.
- cmd/chg .perl_wrapper: header.
- cmd/chg .ksh_wrapper: header.
- cmd/chg .os_wrapper: header.
- cmd/chg .java_wrapper: header.
- cmd/chg perlenv: header.
- cmd/chg edrcman: order of man pages reorganized.
- cmd/chg Linux-ia/cfg2html: new version 2.66.
- cmd/chg Linux/cfg2html: new version 2.66.
- cmd/chg HP-11/cfg2html: new version 5.30.
- cmd/chg HP-11ia/cfg2html: new version 5.30.
- cmd/chg HP-11i/cfg2html: new version 5.30.
- cmd/chg hwinventory: for Linux return SERVERNAME without domain.
- cmd/fix www/report/js/report.js: pad hours with leading zeros when hour is smaller than 10.
- cmd/chg edrcman: cache settings.
- cmd/new hostlistdat2cfg: to generate the hostlist.cfg config file out of a csv structure definition in a hostlist.dat file.
- cmd/fix hostlist: escape dot (.) in groups specified on the command line to ensure correct group resolution. Therefore e.g.: @ACME.ORG is not any longer resolved identically as @ACMECORP.
- cmd/chg hostlist: print the hostgroup prefix (@) in the output of the hostgroup names. Before the hostgroup was printed without the prefix.
- cmd/chg contrib/pwsafe: use vi settings from edrc/etc/exrc for the vi editor in pwsafe.
- cmd/new appendpdf: append a PDF file to another and write concatenated file to stdout or to an output file.
- cmd/new ipcalc: an IP address subnet calculator.
- cmd/new tinydns: a tiny, fast and secure DNS server.
- cmd/new tinydns-data: create the data file for tinydns.
- cmd/new Linux/strace: trace system calls. This command is called if truss is invoked on Linux.
- cmd/new Linux-ia/strace: trace system calls. This command is called if truss is invoked on Linux.
- cmd/new dnscache: DNS cache.
- cmd/new csvcat: clean a csv file or stream for further processing.
- cmd/chg rcmd: use the settings in exrc and prepare the terminal size before starting the vi editor.
- cmd/chg sysvi: prepare the terminal size before starting the vi editor.
- cmd/fix sysvi: correct missing file type resolution when multiple refiles exist which resulted in a "cannot edit" error.
- cmd/chg sysvi: also allow editing of repository files, if there is currently no active file on the system.
- cmd/chg logview: use the settings in exrc and prepare the terminal size before starting the vi editor.
- cmd/chg swvi: use the settings in exrc and prepare the terminal size before starting the vi editor.
- cmd/chg duvi: use the settings in exrc and prepare the terminal size before starting the vi editor.
- cmd/new contrib/edrcupgrade: install and upgrade an WA2L/edrc installation.
- opt/new patchinstall: -m to print the meta-data of the patch only.
- opt/new patchinstall: -l to list the logfiles of the patch only.
- opt/new edrcman: new option -s to display the used \$MANSECT, resolve of the /etc/man_db.config for the MANSECT setting.
- opt/new input: HHH:MM and HHH:MM:SS that allows time inputs, where the hours are not limited to 24 hours as in HH:MM and HH:MM:SS.
- dir/new var/www/report/var/mylogos: to save own factory logos that are not touched by application patch updates.
- lib/new Linux/libs/libext2fs.so.2: used by mc, mcedit, mcview.
- lib/new Linux/libs/libgpm.so.1: used by mc, mcedit, mcview.
- lib/new Linux/libs/libcom_err.so.2.1: used by mc, mcedit, mcview.

1.5.42

Tue Jan 12 00:11:06 CEST 2012 release

- cfg/chg edrcrevision.cfg: new release 1.5.42 (Mon May 23 19:03:12 CEST 2011)
- cfg/chg lots.cfg: remove unresolved options from config file.
- cfg/chg usage.list: add ftps.
- cfg/chg lib/os_wrapper/mc.cfg: ensure that on HP-UX the midnight commander is started instead of the media changer manipulation utility is started.
- man/chg lots.1m: add description of DATA_PURGE_LAG.
- man/chg lots.cfg.4: add description of DATA_PURGE_LAG and remove hint to unresolved options.
- man/chg EDRC.1: add ftps (1).
- man/chg edrcintro.1: add ftps short description.
- man/chg compatibility.1: add ftps.
- man/new ftps: based on README provided by the application.
- man/fix ksh_wrapper.3: correct step 2) description to "install the ksh script in the edrc/lib/ksh/directory"
- cmd/chg edrcman: revert stderr redirection of man command to /dev/null, this caused that on Unix only 1st manpage has been displayed.
- cmd/chg hwinventory: new fields DNSDOMAIN and IPADDRESS.
- cmd/fix maketemp: more restrictive fallback to internal emulation of mktempdir if the system own command does not succeeds.
- cmd/new ftps: secure FTP for SSL connections to the server.
- cmd/chg www/report/cgi/users.remove: support Firefox password manager.
- cmd/chg www/report/cgi/users.passwd: support Firefox password manager.
- cmd/new www/report/cgi/users.modify: modify an existing user account.
- cmd/chg www/report/cgi/users.add: support Firefox password manager.
- cmd/chg www/report/cgi/statistics: sorting order, also show counts of removed users.
- cmd/chg www/report/cgi/cgi.inc: sort list_users by NAME and not by timestamp.
- cmd/chg lots: add DATA_PURGE_LAG functionality.
- cmd/chg lots: add error checking during data purge.
- cmd/chg lots: remove processing of unresolved options.
- cmd/chg lots: print settings of \$LOTS_CONFIGFILE and \$LOTS_IDENTITIES in usage.
- cmd/chg .java_wrapper: support startup links, whose name is not equal to the class name using the edrc/lib/java/<scriptname>.javaopt file.
- cmd/new lib/logcheckd/interface/OVMail: send mails as in HP Openview.
- cmd/fix bin/HP-11ia/bzip2-1.0.5/bin/bunzip2: add symlink to provide the bunzip2 command.
- cmd/chg Linux/cfg2html: new version 2.25.
- cmd/chg Linux-ia/cfg2html: new version 2.25.
- cmd/chg HP-11ia/cfg2html: new version 2.50.
- cmd/chg HP-11i/cfg2html: new version 2.50.
- cmd/chg HP-11/cfg2html: new version 2.50.
- dir/chg lib/lots: remove directory and contents from application.

1.5.41

Mon May 16 18:56:29 CEST 2011 release

- cfg/chg edrcrevision.cfg: new release 1.5.41 (Sat Jan 15 15:38:31 CET 2011)
- cfg/chg os_wrapper/mc.cfg: EXEC_PRIORITY=OS.
- cfg/chg os_wrapper/mcview.cfg: EXEC_PRIORITY=OS.
- cfg/chg os_wrapper/mccedit.cfg: EXEC_PRIORITY=OS.
- cfg/chg osid.dat.WA2L: DISABLE_REGEX option to disable regular expression resolution.
- man/chg osid.3: -e and -h option and EXIT STATUS.
- man/chg osid.cfg.4: description of DISABLE_REGEX.
- man/chg osid.dat.4: description of the regex behaviour in WA2L format.
- man/chg name.1: add EXIT STATUS section.
- man/new osid.probe.3
- man/chg stat.3: -t option.

- man/chg stat.3: -s type option.
- man/chg patchinstall.1m: symlink timestamp update.
- cmd/fix www/report/cgi/statistics: correct statistics output in month December.
- cmd/chg is_osid: print \$EDRC_SCRIPTNAME if -s is not specified.
- cmd/fix logcheckd: correct logcheckd symlink from logcheckd.41 to logcheckd.42.
- cmd/chg HP-11/cfg2html: new version 4.84.
- cmd/chg HP-11ia/cfg2html: new version 4.84.
- cmd/chg lots: save the state of the actions collect, save, lock, purge also to the job file. Data of unsuccessful save actions is not locked and remains in the lock phase until cleared. The new fields COLLECTSTATE, SAVESTATE, LOCKSTATE and PURGESTATE are listed when invoking the -a print_job and -a list_jobs options.
- cmd/chg title: enhance os compatibility.
- cmd/fix whereami: print 'OS not supported message' to stderr.
- cmd/chg name: OS not supported on unknown operating systems.
- cmd/chg HP-11ia/cpio: recompiled without the /usr/local libraries. Version 2.10 is saved as 2.10a to enable patch installation of identical version.
- cmd/chg osid: osid entries can be defined as regular expression if OSID_DAT_FORMAT is set to WA2L and DISABLE_REGEX is not set to True.
- cmd/chg edrc: print copyright to 2011 in banner.
- cmd/chg edrcman: redirect stderr or man command to /dev/null.
- cmd/fix contrib/scripttree: correct handling of symlink-ed menus and menu points.
- cmd/chg pack: filelist is added to the sadm file.
- cmd/chg patchinstall: adjust timestamps of symlinks that are installed. This feature is only supported on operating systems whose kernel supports changing the timestamp of symlinks.
- cmd/chg www/report/js/report.js: formatting of the pop-up menus.
- opt/new osid: -e to list the entry to be added to osid.dat.
- opt/new osid: -h to print usage.
- opt/new stat: -t option to print CSV header.
- opt/chg stat: -t type option to print file type.

1.5.40

Sat Sep 25 02:35:19 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.40 (Fri Sep 24 02:51:11 CEST 2010)
- cfg/chg timezone.dat: Linux-ia added.
- man/chg compatibility.1: screen added.
- man/chg EDRC.1: screen added.
- man/chg edrcintro.1: description of screen.
- man/new screen.1
- cmd/fix logcheckd: avoid df "No such file or directory" error.
- cmd/new screen: screen manager with VT100/ANSI terminal emulation.
- cmd/chg shell: use /bin/bash as interactive shell whenever it is available on the system.
- cmd/fix contrib/scriptdocbook: handle symlinks within menu structure correctly.
- cmd/cor www/report/cgi/customers: html tag alignment.
- cmd/cor www/report/cgi/reports: html tag alignment.

1.5.39

Tue Sep 21 21:49:46 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.39 (Wed Sep 8 19:52:01 CEST 2010)
- cfg/new ssh-keyadd.cfg: config file for ssh-keyadd.
- cfg/new ssh-keyadd.pub: global key file for ssh-keyadd.
- man/chg edrcintro.1: short description of ssh-keygen.
- man/chg EDRC.1: ssh-keyadd added.
- man/chg compatibility.1: ssh-keyadd and sqlite added.
- man/chg apply2file.1: description of -s.
- cmd/new HP-11ia/gethostbyname: compiled for HP-11ia

- cmd/chg Linux/gzip: new version 1.4.
- cmd/chg www/report/cgi/logo.gallery: generate correct table html syntax.
- cmd/chg www/report/cgi/logo.gallery: add copyright and trademark hints.
- cmd/chg www/report/cgi/about: generate correct html syntax.
- cmd/chg www/report/cgi/customers: generate correct list html syntax.
- cmd/chg www/report/cgi/users.add: generate correct span html syntax.
- cmd/chg www/report/cgi/reports: generate correct list html syntax.
- cmd/chg www/report/cgi/reports: resolve en: text of ..du_index file located in report directories and add it as description to the report list.
- cmd/chg www/report/cgi/reports: list only report directories that have at least the permissions 005.
- cmd/chg www/report/cgi/statistics: count only report directories that have at least the permissions 005.
- cmd/chg hwinventory: avoid command hangs, when evaluating the system serial number from the EEPROM using sneep internally on Solaris.
- cmd/chg hwinventory: trim white-spaces out of csv output, remove trailing white-spaces from record output.
- cmd/cor crfile: units in usage.
- cmd/new ssh-keyadd: add SSH public keys to a users authorized_keys file(s) and modify the SSH daemon configuration file to setup the needed ssh trusts to connect.
- opt/new apply2file: -s to save a file prior of applying the data stream.

1.5.38

Wed Sep 8 10:57:33 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.38 (Tue Aug 31 21:46:05 CEST 2010)
- cfg/chg lib/usage.list: add osid.probe.
- man/new unczp.1
- man/chg EDRC.1: add unczp.1, unrpm.1.
- man/chg edrcintro.1: short description of unczp.
- man/chg unczg.1: add unczp and unrpm to SEE ALSO section.
- man/chg untgz.1: add unczp and unrpm to SEE ALSO section.
- man/new unrpm.1
- cmd/chg HP-11/cpio-2.10: adjust library binding in binary from static to dynamic.
- cmd/chg contrib/scriptdocbook: table in title page to 100% of page width.
- cmd/chg contrib/scriptdocbook: use apprevision to resolve WA2L/edrc version.
- cmd/chg contrib/scriptdocbook: unset environment variables, that influence the creation of the scriptdocbook archive file.
- cmd/chg contrib/scriptpack: unset environment variables, that influence the creation of the scriptdocbook archive file.
- cmd/chg lscomp: unset environment variables, that influence the listing of the compressed archive files.
- cmd/chg llcomp: unset environment variables, that influence the listing of the compressed archive files.
- cmd/chg catcomp: unset environment variables, that influence the display of the compressed archive files.
- cmd/new apprevision: return version of WA2L/edrc, the output is identical to the sat -V respectively edrc -V call, but faster.
- cmd/chg www/report/cgi/about: use apprevision to resolve WA2L/edrc version.
- cmd/chg www/report/cgi/links: allow '---' to add a separator between references.
- cmd/chg hwinventory: use apprevision to resolve WA2L/edrc version.
- cmd/fix histlist: correct begin timestamp output (???-??-??) when history file does not exist.
- cmd/fix histlist: eliminate empty lines.
- cmd/fix sysdiff: no empty file name output.
- cmd/new unczp: extract files of a zipped cpio file.
- cmd/new unrpm: extract files of a rpm file.
- cmd/chg patchinstall: umask setting for HP-* osid's to ensure correct directory permissions.
- cmd/chg sysconfig: umask setting for HP-* osid's to ensure correct directory permissions.

1.5.37

Tue Aug 31 13:57:59 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.37 (Sun Aug 15 13:45:29 CEST 2010)
- cfg/new osid.dat.WA2L: new osid Solaris-x86.
- cfg/fix os_wrapper/pl.cfg: make it sh compliant.
- cfg/chg osid.cfg: setting PROBE_BINARY to set the use of the osid.probe command.
- man/chg osid.3: description of the -n and -s options.
- man/chg compatibility.1: Solaris-x86 added.
- man/new man/Solaris-x86: man pages.
- man/chg tthttpd.3: remove manpage that explains -now option.
- man/chg edrcrevision.1: description of osid.probe.
- man/chg osid.cfg.4: description of PROBE_BINARY.
- man/chg stat.3: description of atimel, ctimel, mtimel and -f - option.
- cmd/chg tthttpd report: minor improvements, version 1.0.01.
- cmd/new many: compiled Solaris-x86 commands.
- cmd/chg: osid: probes if the OSID is correct to execute binaries. This is done via lib/OSID/osid.probe.
- cmd/new: lib/<OSID>/osid.probe: probe if it is possible to start a binary related to the chosen osid.
- cmd/opt tthttpd: remove -now option again
- cmd/fix contrib/newscripttree: _title is no longer empty.
- cmd/chg edrcinit: support of chkconfig when calling 'edrcinit rcinstall' on Linux.
- cmd/chg stat: can now also print atime, mtime and ctime in local time representation, not only as timer since the epoch. The all field also includes the local time representation fields.
- cmd/chg stat: can now read a list of files provided thru stdin.
- cmd/chg histlist: significant performance increase (real 1m47.090s -> real 0m2.521s). The jumping point is no longer needed.
- cmd/fix hwinventory: avoid error outputs, check for sneep on Solaris, if serial number is 0, return empty serial number.
- opt/new osid: -p and -s to print the definitions.

1.5.36

Fri Aug 13 13:46:32 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.36 (Tue Aug 10 23:05:00 CEST 2010)
- cfg/new osid.dat.WA2L: new osid Linux-ia.
- cfg/chg kshrc: display .WA2L files in the color of config files.
- man/chg tthttpd.3: only Linux shows the -now option in the man page.
- man/chg compatibility.1: Linux-ia added.
- cmd/chg tthttpd report: major improvements, version 1.0.00.
- cmd/new many: compiled Linux-ia commands.

1.5.35

Mon Aug 9 21:08:35 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.35 (Mon Jul 19 23:13:01 CEST 2010)
- man/chg contrib.edrc.1m: description of contrib.newscripttree.
- man/chg tthttpd.3: copy to man/Solaris, man/Linux.
- cmd/fix contrib/scriptdocbook: mail subject, mailing of the created file.
- cmd/chg contrib/scriptdocbook: user dialog.
- cmd/fix contrib/scriptpack: mail subject, mailing of the created file.
- cmd/chg contrib/scriptdocbook: user dialog.
- cmd/chg edrc: cleaner output of edrcperm output.
- cmd/new contrib/newscripttree: create a new script tree based on a template.
- cmd/chg syscp, sysvi, sysdiff: can now handle symlinks, directories and files.
- cmd/chg tthttpd report: improvements, version 0.0.02.
- cmd/chg bin/<OSID>/cfg2html: preparation to dynamically create a temporary systeminfo file containing the output of the hwinventory command.

1.5.34

Mon Jul 5 20:33:57 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.34 (Wed May 5 14:52:06 CEST 2010)

- cfg/new timezone.dat: map file for timezone setting on Linux.
- man/new banner.3
- man/chg pstree.1: -p option description.
- man/chg contrib.edrc.1m: enhance description of contrib.scriptpack.1m.
- man/chg contrib.edrc.1m: enhance description of contrib.scriptdocbook.1m.
- man/chg timezone.3: add -v and timezone.dat description.
- man/chg tthtpd.3: move to man/all.
- man/chg tthtpd.3: fix: nosymlinkcheck/symlinkcheck configfile option to nosymlink/symlink.
- man/chg tthtpd.3: description of the -now (nowworldreadable) feature.
- man/chg fssum.1: SYNOPSIS.
- cmd/fix lots: setting of the access time for files.
- cmd/fix lots: output of 'Configfile:' when \$LOTS_CONFIGFILE environment variable is set.
- cmd/chg contrib/sessions: forced kill of edrc sessions and cleanup of remaining session sub-processes when cleaning up inactive sessions.
- cmd/chg contrib/scriptdocbook: allow ~user in FILEPATH of scriptdocbook.bib.
- cmd/chg contrib/scriptdocbook: includes a (printable) 'Table Of Contents' page.
- cmd/chg contrib/scriptdocbook: formatting of 'HEADER', 'BIBLIOGRAPHY' and 'TABLE OF CONTENTS' pages adjusted.
- cmd/chg Linux/cfg2html: version 1.74.
- cmd/chg HP-11*/cfg2html: version 4.65.
- cmd/chg sysvi,sysdiff,syscp,svi: ensure that only the basic (without acl) permissions are compared.
- cmd/chg sysconfig: ensure that only the basic (without acl) permissions are compared.
- cmd/chg contrib/scriptpack: allow to choose between saving or mailing the generated script tree package.
- cmd/chg contrib/scriptdocbook: allow to choose between saving or mailing the generated script tree package.
- cmd/chg tsiz: set maximal columns for vi on SunOS (Solaris) versions prior to Solaris 10.
- cmd/new tzdump: dump contents of a zoneinfo file.
- cmd/chg timezone: has a map file (timezone.dat) to correct TZ on Linux if needed.
- cmd/chg fssum: no longer use cmviewcl to resolve packages.
- cmd/opt tthtpd: -now to disable the checking if the served file is world-readable.
- cmd/cor shell: message, when OS-ID is unknown points to the correct file, now.
- cmd/chg contrib/pwsafe: when displaying passwords using the s) or h) options, print the output directly to /dev/tty to ensure not logging the output. Be aware, that the pwsafe.nolog is still needed to ensure, that e) is not logged.
- cmd/fix fssum: argument parsing, now -t [-v] is working.
- cmd/chg fssum: if FREE + USED > KBYTES, the KBYTES output is FREE + USED. This corrects the behaviour as seen on Solaris, where KBYTES is reported as 0, when the file system is on ZFS.
- cmd/chg fssum: also report EXCLUDE lines when running in verbose (-v) mode.
- cmd/cor fssum: usage message output.
- opt/new pstree: -p to print a process-id list of the process tree only.

1.5.33

Thu Mar 25 22:52:08 CET 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.33 (Wed Jan 27 22:53:10 CET 2010)
- man/fix lots.cfg.4: SYNOPSIS.
- man/chg contrib.edrc.1m: contrib.scriptdocbook description.
- man/cor contrib.edrc.1m: grammar in contrib.scriptpack(1m) short description.
- man/new Linux/dmidecode.3: for dmidecode version 2.9.
- man/chg edrcintro.1: description of dmidecode.
- man/chg edrcintro.1: description of scriptrevision.
- man/chg compatibility.1: dmidecode added.
- man/chg compatibility.1: scriptrevision added.
- man/chg EDRC.1: dmidecode added.
- man/chg EDRC.1: scriptrevision added.

- man/new os_wrapper.1
- man/chg ksh_wrapper: SYNOPSIS precised.
- man/chg perl_wrapper: SYNOPSIS precised.
- cmd/chg contrib/scriptdocbook: a BIBLIOGRAPHY section can be added to the ScriptDocBook when a scriptdocbook.bib file is created in the related \$EDRC_ENTRY_DIR directory.
- cmd/chg contrib/scriptpack: only ask for removal of pwsafe.dat if it exists in the related recovery script tree.
- cmd/fix contrib/scriptpack: ensure correct output when copying script files.
- cmd/chg HP-11/cpio: new version 2.10 .
- cmd/chg HP-11i/cpio: new version 2.10 .
- cmd/fix scriptdocbook: bug that wrote to /devnull instead of /dev/null.
- cmd/fix is_up: correct intern handling of return code of ping which caused that is_up never checked the ssh port when ping is not allowed thru firewalls.
- cmd/new HP-11ia/portscan: compiled.
- cmd/fix: lots: internal fix in get_diskusagestats(). This bug was not visible to the user.
- cmd/chg lots: use scriptrevision command internally.
- cmd/new Linux/dmidecode: DMI table decoder.
- cmd/chg .os_wrapper: allows configuration for each wrapped command.
- cmd/fix .os_wrapper: argument passing to wrapped commands allows strings containing white spaces.
- cmd/fix .perl_wrapper: argument passing to wrapped perl scripts allows strings containing white spaces.
- cmd/chg Linux/collectd: version 4.9.1.
- cmd/new scriptrevision: return revision of a (script) file having the [##] revision notation as resolved also by the revision command.
- cmd/chg server_environment: cache meta data format changed, the md5sum command is no longer required.
- cmd/chg hostlist: cache meta data format changed, the md5sum command is no longer required.
- cmd/chg du_report: exclude /zones /proc /tmp file systems depending on the OSID.
- cmd/chg sw_report: exclude /zones /proc /tmp file systems depending on the OSID.
- cmd/new banner: to print login banner.
- cmd/new hwinventory: to print the minimal set of needed asset management information.

1.5.32

Fri Jan 27 04:40:06 CEST 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.32 (Thu Jan 21 21:28:24 CET 2010)
- cfg/chg kshrc: new alias histlist to print a list of history files including the timerange.
- man/cor lots.1m: SYNOPSIS, OPTIONS, typing errors.
- man/cor directories.3: formatting.
- man/chg compatibility.1: mediawiki and CMS:Mediawiki compatible to Solaris and HP-11ia.
- man/chg compatibility.1: fcreate.
- man/chg lots.1m: description of all job properties printed when using -a print_job.
- man/chg EDRC.1: add histlist (3).
- man/chg EDRC.1: add fcreate (3).
- man/chg compatibility.1: add histlist (3).
- man/chg edrcintro.1: add histlist (3).
- man/chg edrcintro.1: add fcreate (3).
- man/chg lots.1m: BUGS added.
- cmd/chg CMS:Mediawiki: compiled for Solaris and HP-11ia.
- cmd/fix directories: eliminate empty line output.
- cmd/fix sysconfig: avoid 'sed: -e expression #1, char 29: Invalid range end' error while writing settings.
- cmd/chg edrc: write logheader to SHELL logfiles.
- cmd/chg lots: print total source size when invoking -a list_jobs.
- cmd/chg lots: print PURGETIMEDAT when invoking -a list_jobs.
- cmd/chg lots: when issuing -a print_logtail the screen is fully filled with the logfile.
- cmd/chg lots: LOCK_TIMEOUT default is now 22 hours.

- cmd/chg lots: print 'collect disk usage statistics' when collecting it to avoid the impression that the command hangs.
- cmd/chg lots: perform disk usage collection at maximum once per session, not once per action.
- cmd/chg lots: all GByte outputs have 5 digits after the comma.
- cmd/chg shell: shell history file name when invoking shell outside edrc, save shell history in SHELL log-file when invoking from within edrc.
- cmd/new histlist: print a list of history files.
- cmd/chg usage: avoid repeating outputs of identical usages when in the definition file tabs where used.
- cmd/new fcreate: create a file if it does not already exist.

1.5.31

Sun Jan 17 18:47:41 CET 2010 release

- cfg/chg edrcrevision.cfg: new release 1.5.31 (Fri Aug 14 15:23:25 CEST 2009)
- cfg/chg pack/edrc/shar.header: Copyright 2003-2010
- cfg/chg bin/index.html.template: Copyright 2003-2010
- cfg/chg contrib/edrc/license: Copyright 2003-2010
- cfg/chg contrib/edrc/scriptdocbook: Copyright 2003-2010
- cfg/chg contrib/edrc/scriptpack: Copyright 2003-2010
- cfg/chg logcheckd.cfg: using the options REPORT_MAIL_CC and REPORT_MAIL_BCC the reports can also be mailed to carbon copy and blind carbon copy recipients.
- man/new consolidate.1
- man/new stat.1
- man/new CMS::MediaWiki.3
- man/chg compatibility.1: CMS::MediaWiki.
- man/chg perl_modules.3: CMS::MediaWiki.
- man/chg edrcintro.1: SEE ALSO section updated with missing Perl modules.
- man/chg EDRC.1: add mediawiki.3.
- man/chg edrcintro: short introduction to mediawiki command.
- man/chg vsdfml.3: description of MEDIAWIKI format option.
- man/chg contrib.edrc.1m: description of pwsafe.
- man/chg contrib.edrc.1m: description of checklist.
- man/new lots.1m
- man/new schedule.dat.4
- man/chg EDRC.1: lots, lots.cfg, datalist.dat, schedule.dat, volume.dat.
- man/chg edrcintro.1: lots description, SEE ALSO.
- man/new volume.dat.4
- man/new datalist.dat.4
- man/new lots.cfg.4
- man/chg logcheckd.cfg.4: description of REPORT_MAIL_CC and REPORT_MAIL_BCC.
- man/chg logcheckd.1m: description of REPORT_MAIL_CC and REPORT_MAIL_BCC.
- man/new mediawiki.3
- man/chg stat.3: description of -a all option.
- man/new pwcrypt.1
- man/new revision.1
- man/chg name.1: formatting, SEE ALSO.
- man/new symlink.1
- man/new directories.3
- man/chg sparse.3: SEE ALSO.
- man/cor edrcintro.1: description example of stat.
- man/chg busy.1: description of \$BUSY_TURNING_INTERVAL.
- cmd/chg contrib/scriptdocbook: start command in index.html.
- cmd/chg contrib/scriptdocbook: print filename at end of zipping it.
- cmd/chg HP-11ia/cpio: version 2.10.
- cmd/fix contrib/scriptdocbook: unset suffix error when 'set_dir' is denied.

- cmd/new CMS::MediaWiki: Perl extension for creating, reading and updating MediaWiki pages.
- cmd/new mediawiki: get/put page from/to MediaWiki.
- cmd/fix logrotate: fix "logrotate[35]: approot: not found" error.
- cmd/chg vsdfml: new MEDIAWIKI format output.
- cmd/fix vsdfml: escape backslashes in input to ensure proper backslash output.
- cmd/fix contrib/doc: leading blanks of output examples (O: txt) are no longer removed.
- cmd/new contrib/pwsafe: interactive password safe with the possibility to encrypt the password data file.
- cmd/chg contrib/calc: the command is denied from startup if shell is listed in DENY_LIST, functionality contrib.calc.shell does not exist any more.
- cmd/fix edrc: avoid "Terminated ..." error.
- cmd/fix edrc: avoid "find : cannot find current directory" error.
- cmd/chg lsmv: first column is dynamic, now.
- cmd/chg lscp: first column is dynamic, now.
- cmd/chg contrib/doc: redirect potential man error messages to /dev/null.
- cmd/fix contrib/doc: escape backslash in input to ensure the backslash in the output.
- cmd/chg contrib/doc: provide the .VB / .VE makros.
- cmd/chg contrib/doc: <pre>, </pre> markup supported, now.
- cmd/new contrib/checklist: print a checklist that can be used to track disaster recovery progress based on the recovery script tree.
- cmd/new contrib/scriptpack: pack the script tree to a package.
- cmd/new expand file list with overlying directories.
- cmd/new HP-11ia/info:
- cmd/chg pack: include overlying directories also into patch.
- cmd/chg Linux/gzip: version 1.3.12 -> 1.3.14.
- cmd/chg edrc: copyright 2010
- cmd/chg llcomp: print zip compression comment, too.
- cmd/fix contrib/doc: avoid "bad string" error.
- cmd/fix filesize: avoid -b usage error.
- cmd/fix bin/*/cfg2html: ensure, that the commands in ./plugins are found by cfg2html.
- cmd/chg HP-11*/cfg2html: version 4.53.
- cmd/chg bin/<OSID>/gzip: is now a symbolic link to the command with the name gzip-<VERSION>.
- cmd/chg bin/<OSID>/cpio: is now a symbolic link to the command with the name cpio-<VERSION>.
- cmd/chg logcheckd: reports can now be sent to CC and BCC recipients, too.
- cmd/chg csv2worksheet: use the new POI Java library poi-3.6-20091214.jar.
- cmd/chg stat: new -a all stat field to print all stat information in one query as a csv output.
- cmd/chg busy: using \$BUSY_TURNING_INTERVAL the speed of the rotating slash can be set, now.
- opt/new filedist: -t hostlist.
- opt/new stat: -s all.

1.5.30

Thu Aug 13 14:28:05 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.30 (Mon Aug 10 21:28:29 CEST 2009)
- cfg/chg bin/usage.list: add lspm.
- man/chg mail_file.1: change cole to carbon.
- man/new File::Tail.3: Perl extension for reading from continuously updated files.
- cmd/chg contrib/scriptdocbook: query ScriptDocBook suffix.
- cmd/chg fssum: the system in the fssum.cfg file can be a regular expression, too.
- cmd/chg mail_file: change cole to carbon.
- cmd/chg streamcat: reduce CPU consumption (this is only an intermediate solution).
- cmd/new lib/perl/pm: new module File::Tail.

1.5.29

Thu Aug 6 13:36:53 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.29 (Sat Jul 18 16:08:53 CEST 2009)
- cfg/chg edrcman.cfg: accept also white space separated MAN_PATH setting. This enables to use wild-cards.

- cfg/chg kshrc: ensure that the manvi alias starts always the real command in a ksh or clone.
- man/new MediaWiki::API.3
- man/chg EDRC.1: add sortc (3).
- man/chg EDRC.1: add lspm (1).
- man/chg edrcintro.1: short description for sortc.
- man/chg edrcintro.1: short description for lspm.
- man/chg compatibility: add sortc.
- man/chg fssum.cfg.4: ensure that the complete file format string is printed on Solaris, too.
- man/chg fssum.cfg.4: description behaviour when specifying "h:localhost:".
- man/chg logcheckd.cfg.4: ensure that the complete file format string is printed on Solaris, too.
- man/new lspm.1: list installed Perl modules.
- man/chg perl_modules.3: add reference to lspm.
- cmd/new lib/perl/pm: new module Unicode:String.
- cmd/new lib/perl/pm: new module MediaWiki:API and required dependencies.
- cmd/chg pack: more verbosity on package creation to give the user the visibility that the package creation is progressing.
- cmd/chg rcmd: do not set the LC_ALL environment variable internally. This influenced certain commands started with rcmd.
- cmd/chg edrcman: also compute white space separated MAN_PATH in edrcman.cfg.
- cmd/chg cltrash logtail passwdcombine edrc cronhandler edrcinit job logcheckd passwdsyncd patchinstall checkopt collectd fmatch hostlist ini.bash listtemp pkg_hostname pscount resolve_targetlist role_option server_environment thttpd scriptextract bwcreate cmmon cmswitch consolidate du_report edrcenv filedist filelink fssum ipsort locations lscp lsmv makeuser name pack passwdsort pstree rcmd revision sw_report sysconfig usage user_info whoisin ypxfr_all: do not set the LC_ALL environment variable internally to avoid unwanted side effects.
- cmd/new sortc: sort date in the "C" locale.
- cmd/chg DocBrowser: displays a loading image while loading the main browser window.
- cmd/chg contrib/scriptdocbook: displays a loading image while loading the main browser window.
- cmd/new streamcat: continuous output of a stream received via stdin to stdout.
- cmd/chg edrc: avoid command hanging as observed on on certain legacy Linux systems.
- cmd/chg fssum: when localhost as system is specified in the fssum.cfg file it will be replaced with the hostname in the report.
- cmd/fix fssum: report sum calculation, when issuing fssum without options, does not cause a value overflow any more.
- cmd/new lspm: list installed Perl modules.

1.5.28

Tue Jul 14 15:01:59 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.28 (Fri Jul 10 11:16:54 CEST 2009)
- cfg/chg kshrc: load working variables at the very last stage to ensure that the important parts of the kshrc file are processed independent from disturbing variable content.
- man/chg sav.1: description of handling of files located in special operating system directories.
- cmd/chg sav: automatically save files residing in additional directories then /etc/rc.config.d in a \$TODAY subdir.
- cmd/fix edrc: error message when invoking the 'log' edrc command.
- cmd/fix edrc: indefinite loop when distributing to a single system.

1.5.27

Fri Jul 10 10:44:55 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.27 (Tue Jun 9 12:40:41 CEST 2009)
- cfg/chg logcheckd/WA2L.css: printing definitions.
- cfg/chg kshrc: start of editors (vi, vim, view) does no longer influence the \$COLUMNS, \$LINES environment variables in the shell.
- cfg/chg shell.cfg: EDRC_SW_BASEDIR replaced with EDRC_SW_PATH.
- cfg/chg shell.cfg: SHELL_BANNER to be displayed when shell is called outside of edrc.

- man/chg csv2worksheet.3: description of -s option, EXAMPLES text formatting changed.
- man/chg edrcintro.1: short description of consolidate.
- man/chg EDRC.1: consolidate.
- man/chg compatibility.1: consolidate.
- man/chg logcheckd.style.4: description of additional CSS classes.
- man/chg logcheckd.1m: description of <number>.excluded files.
- cmd/chg csv2worksheet: possibility to explicitly load data as text using the -c cols option.
- cmd/new consolidate: to consolidate structured unix file data.
- cmd/fix logcheckd/interface/CAUnicenter: remove trailing pipe character from log entry.
- cmd/fix logcheckd: preserve quotes when relaying log entries to the interfaces.
- cmd/chg logcheckd: additional CSS classes introduced to control the layout more fine-grained.
- cmd/chg logcheckd: no longer represent EXCLUDE hits as <CR> in collect directory.
- cmd/fix edrc: bug on parallel distributions when having a DIST_TYPE=parallel:xy where xy is less than the total number of distribution targets.
- cmd/fix edrc: messed up logfile on parallel distribution.
- cmd/chg tsize: new option 'vi' to influence max. columns mainly on HP-UX when initializing terminals prior to vi, vim, view start.
- cmd/chg shell: connection speed enhanced, ~/.eshell/cache no longer used.
- cmd/chg log: now allowed to write to named pipes, too.
- cmd/chg logview: avoid vim warning 'output is not a tty'.

1.5.26

Fri Jun 12 09:54:10 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.26 (Tue Jun 9 12:40:41 CEST 2009)
- cfg/chg logcheckd/WA2L.css: div.timing -> span.timing, div.separator -> span.separator.
- cfg/chg logcheckd/WA2L.css: rows in Overview table are highlighted on mouse hover.
- cfg/chg kshrc: on HP-* set terminal width to a maximum of 200 characters when starting vi, vim, view to avoid "Terminal too wide" error.
- man/chg mail_file.cfg.4: description of MAIL_HUB and MAIL_HUB_USER.
- man/chg mail_file.1: description of mail hub functionality.
- man/chg lsof.1: for version 4.82.
- man/chg lsof: version 4.82 for Linux, HP-11i, HP-11, Solaris.
- man/new top.1: for Solaris.
- man/chg edrcintro.1: top.
- man/chg EDRC.1: top.
- man/chg compatibility.1: top.
- cmd/chg logcheckd: logfile content HTML in LogCheck report to support one line copy-paste from report to mail or text editor without having breaks after the timestamp separator.
- cmd/chg logcheckd: title on all fields.
- cmd/chg mail_file: mail via hub.
- cmd/new top: for Solaris.
- cmd/chg edrc: on HP-* set terminal width to a maximum of 200 characters when starting vi, vim, view to avoid "Terminal too wide" error.

1.5.25

Tue Jun 9 10:15:45 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.25 (Fri Jun 5 21:06:15 CEST 2009)
- cfg/chg patchinstall.cfg: new settings BACKUP_PATCHED_FILES_GENERATIONS, BACKUP_CORPSES_GENERATIONS to define the number of backup generations kept while purging old backup copies.
- cfg/chg edrc.cfg: new setting BACKUP_GENERATIONS to specify the number of kept script backups when using the 'distribute' command.
- man/chg patchinstall.1m: description of the purging steps, change of the example output.
- man/chg patchinstall.cfg.4: description of BACKUP_PATCHED_FILES_GENERATIONS and BACKUP_CORPSES_GENERATIONS options.

- man/chg edrc.cfg.4: description of BACKUP_GENERATION option.
- man/chg lsof: version 4.81 for HP-11 and HP-11i.
- cmd/fix passwdsyncd: error prone resolving of remote home.
- cmd/fix shell: egrep error message.
- cmd/new cfg2html: version 4.41 for HP-11, HP-11ia.
- cmd/new msmtmp: an SMTP client.
- cmd/chg patchinstall: can now purge old backups of patched files and corpses.
- cmd/chg edrcinit: review.
- cmd/chg scriptextract: can now purge old backups of distributed scripts.
- cmd/fix pack: timestamp is always in GMT.
- cmd/chg lsof: version 4.81 for HP-11 and HP-11i.

1.5.24

Fri Jun 5 11:06:37 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.24 (Tue Jun 2 18:54:49 CEST 2009)
- cfg/chg logcheckd.cfg: ALERT_MAIL_TO and ALERT_MAIL_SECTION to handle system personnel information.
- man/chg logcheckd.cfg.4: description of settings ALERT_MAIL_TO and ALERT_MAIL_SECTION.
- cmd/fix shell: avoid ~user/.eshell/cache cannot create error.
- cmd/chg logcheckd: can now send an alert mail on analysis suspension, resumption and on non-interactive report creation to inform system management personnel about the filesystem shortage.
- cmd/fix sparse: syntax error.
- cmd/fix contrib/clonemenu: does not report a clone target menu as not empty if only files exist in the .sav directory.

1.5.23

Mon May 16 18:56:29 CEST 2011 release

- cfg/chg edrcrevision.cfg: new release 1.5.23 (Fri May 29 19:48:36 CEST 2009)
- cfg/chg WA2L.css: new tags to allow current log count marking.
- man/chg logcheckd.1m: add pattern file row numbers to example 1), explain the criteria to avoid the filesystem fill up.
- man/chg logcheckd.1m: description of -a defined and -a monitored options.
- man/chg edrcintro.1: description of edrc/.ssh directory in section FILES.
- man/chg logcheckd.style.4: new tags to allow current log count marking.
- cmd/chg logcheckd: when verifying a pattern file using the -a verify option, the row number of the patching line in the patternfile is printed left to the matched log file entry.
- cmd/chg logcheckd: current counts that are bigger and equal to zero can be marked, now.
- cmd/chg install: set ownership of edrc/.ssh based on input.
- cmd/chg: scriptdocbook: HTML code enhanced.
- cmd/cor: filesize: unit shortcuts in usage message output.
- cmd/cor: freespace: unit shortcuts in usage message output.
- cmd/new: logcheckd/interface/CAUnicenter: interface for CAUnicenter Log agent.
- opt/new logcheckd: -a defined / -a monitored to list the defined / currently monitored logfiles.
- dir/new edrc/.ssh: *empty* directory to ensure ssh execution in some circumstances (do *not* save ssh files into this directory).

1.5.22

Mon May 16 18:56:29 CEST 2011 release

- cfg/chg edrcrevision.cfg: new release 1.5.22 (Wed May 20 09:10:38 CEST 2009)
- cfg/chg logcheckd.cfg: INTERFACES, INTERFACECONFIGDIR to configure the external interfaces.
- man/cor contrib.1m: example 2 code.
- man/chg contrib.edrc.1m: description of contrib.scriptdocbook(1m).
- man/chg logcheckd.cfg.4: description of INTERFACES, INTERFACECONFIGDIR settings.
- man/chg logcheckd.1m: description of interfaces and iconfig directories.
- man/chg EDRC.1: add edrcinit.1m, edrcinit.cfg.4.
- man/chg edrcintro.1: add edrcinit.1m.

- cmd/chg httpd: internal function renaming to enhance compatibility.
- cmd/chg hostlist: output formatting of hostlists when using the -p option.
- cmd/chg contrib/doc: troff formatting of doc tags O: and F:.
- cmd/chg contrib/scriptdocbook: active trunks are represented, now.
- cmd/chg contrib/scriptdocbook: new section "ICON INDEX" in main page.
- cmd/chg contrib/scriptdocbook: man page contrib.edrc.1m also added to ScriptDocBook.
- cmd/chg logcheckd: duration output in non-engineering format.
- cmd/chg logcheckd: external interface support.
- cmd/fix watchdog: print list on Solaris did not return a list, even when watchdogs were running.
- cmd/fix watchdog: -L did not succeed on Solaris.
- cmd/fix watchdog: formatting of output table when issuing the -l option.
- cmd/chg pslist: avoid egrep errors (as "egrep: \$ anchor not at end of pattern.") on wrong patterns.
- cmd/chg pslist: avoid "zonename: not found" error on Solaris.
- cmd/fix mail_file: avoid "could not open /dev/kbd" and related errors as seen on Solaris.
- cmd/fix mail_file: option parsing which caused usage message on certain subject texts.
- cmd/chg shell: has a simple cache of remote WA2L/edrc installation homes.
- cmd/new edrcinit: handle WA2L/edrc services.
- cmd/chg untgz uncgz lscomp llcomp catcomp whatis apropos yesterday tomorrow today remote_shell
remote_copy removetemp maketemp listtemp ini.sh ini.ksh ini.csh ini.bash freespace filesize
edrcroot aproot: react on EDRC_DEBUG_<COMMAND> on all hardlinked variants of the
command.
- opt/chg logcheckd: -a state to -a status.

1.5.21

Tue May 19 19:45:11 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.21 (Fri May 8 19:17:41 CEST 2009)
- cfg/fix kshrc: correct output of a, b, c, d shell commands.
- cfg/chg logcheckd.cfg: new setting ANALYSIS_SUSPENSION.
- cfg/chg logcheckd.cfg: new setting MIN_FREESPACE.
- man/chg edrc.1m: description of the behaviour of direct trunks.
- man/chg contrib.edrc.1m: add contrib.scriptdocbook.
- man/chg logcheckd.cfg.4: CHECK_INTERVAL description.
- man/chg edrcman.1: description of -w option, SEE ALSO, FILES.
- man/chg shell.1: options.
- man/chg logcheckd.pattern.4: description that identical section names can appear multiple times in a pattern file.
- man/chg logcheckd.1m: description of analysis suspension.
- man/chg regexintro.4: example for the usage of "*".
- man/new man2html.3
- man/chg vsdfml.3: explain format option.
- cmd/chg contrib/scripttree: compress titles as done in edrc.
- cmd/chg contrib/doc: internal improvement to be able to use the command in an other contributed command.
- cmd/chg edrc: if on direct trunks a list of targets is given, probe the whole list and establish the trunk to the first target that is up.
- cmd/fix edrc: report if a trunk specification is bugous. This has been done in previous versions, but the exit code was overwritten by a later update.
- cmd/new contrib/scriptdocbook: create a DocBrowser book for a recovery script tree.
- cmd/chg DocBrowser: allow to resize the menu panel.
- cmd/fix logcheckd: usage display when -a verify patternfile LEVEL is used.
- cmd/fix logcheckd: setting of CHECK_INTERVAL default.
- cmd/fix logcheckd: locking error (Solaris problem only).
- cmd/fix is_writeable: when probing file access, remove tailing /-es of file/dir specified to be more os compatible.

- cmd/fix sav: "timezone: command not found" error.
- cmd/chg edrcman: has now a config file where extensions to the MANPATH can be specified.
- cmd/chg contrib/logs: extra line feed before --- line to avoid messy line breaks on logfiles not having a line feed as the last character.
- cmd/chg contrib/logs: additional selection "(t)his" to redisplay the currently selected logfile.
- cmd/chg cronhandler: internal improvement to enhance compatibility when handling the lock file.
- cmd/chg watchdog: internal improvement to enhance compatibility when handling the lock file.
- cmd/chg passwdsyncd: internal improvement to enhance compatibility when handling the lock file.
- cmd/chg passwdsyncd_apply: internal improvement to enhance compatibility when handling the lock file.
- cmd/chg logcheckd: internal improvement to enhance compatibility when handling the lock file.
- cmd/chg logcheckd: avoid file system fill ups caused by the daemon.
- cmd/fix job: release number.
- cmd/chg shell: possibility to connect to remote hosts via edrc mechanism and start the shell on the remote system.
- cmd/chg shell: separate shell history per tty session.
- cmd/chg pscount: more tolerant CSV input processing.
- cmd/new man2html: convert troff to HTML.
- opt/new logcheckd: -a state , to check if a daemon is running.
- opt/chg hostlist: -p, to print the configures host groups.
- opt/new vsdfml: format, to format files into man-troff.

1.5.20

Fri May 8 00:35:15 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.20 (Thu May 7 23:15:17 CEST 2009)
- man/chg shlib.3: FILES section.
- man/chg perl_wrapper.3: SEE ALSO.
- cmd/fix edrc: error while distributing in parallel mode.
- cmd/chg edrc: menu title output compressed.
- cmd/chg pscount: sort the output of the uncounted process list.
- cmd/fix hostlist: do not hang when cache files are empty.
- cmd/chg lib/perl/pm: compiled all Perl modules for i586-linux-thread-multi.
- cmd/chg shlib: standard library path for Solaris and Linux.
- cmd/chg .os_wrapper: shared library environment variable exporting.
- cmd/chg .perl_wrapper: library environment variable exporting.
- cmd/fix fssum: EXCLUDE handling.

1.5.19

Mon May 4 17:54:49 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.19 (Thu Apr 30 15:57:28 CEST 2009)
- man/cor edrcsetup.1m: ssh-keygen command.
- man/new fssum.cfg.4
- man/new shlib.3
- man/new psjoin.3
- cmd/new psjoin: concatenate PostScript files.
- cmd/chg nano: default tabstop is 4 now.
- cmd/chg rnano: default tabstop is 4 now.
- cmd/fix edrc: remote home resolution while executing the "distribute" command.
- cmd/chg shlib: set an os standard path for Linux, too.

1.5.18

Thu Apr 30 10:12:48 CEST 2009 release

- cfg/chg edrcrevision.cfg: new release 1.5.18 (Fri Dec 12 20:09:59 MET 2008)
- cfg/chg timezone.cfg: comments in file changed.
- cfg/chg shell.cfg: use is_existing to resolve the software location.
- cfg/chg kshrc: print of \$MANPATH now based on edrcman -w output.
- cfg/chg thttpd.cfg: renamed to thttpd.doc.cfg.

- cfg/new thttpd.vsdms.cfg: httpd configuration for the vsdms (under development).
- cfg/new thttpd.base.cfg: httpd configuration for the base httpd configuration (under development).
- cfg/new termcap: terminal capability definition for commands relying on termcap instead of terminfo.
- cfg/chg *.cfg: VAR='homedir'/dir to VAR='homedir/dir' to profit from the default when the user ACME does not exist on a system. This enhances bootstrap start after installation.
- cfg/chg excrc: remove ignorecase option.
- man/new compatibility.1: command compatibility overview.
- man/chg shell.1: description of termsize, vi, view and vim.
- man/chg sqlite.1: hint how to read the HTML documentation.
- man/cor EDRC.1: section of sysconfig.cfg.
- man/chg EDRC.1: reference to the *.zip archive in edrc/doc containing all documentation files.
- man/chg edrcintro.1: section of sysconfig.cfg.
- man/chg edrcintro.1: examples for select_columns, print_list, print_index.
- man/chg edrcintro.3: add perlenv short description.
- man/new print_list.3
- man/new print_header.3
- man/new print_index.3
- man/new timezone.cfg.3
- man/new timezone.cfg.4
- man/new whereami.1
- man/new logview.1
- man/new logtail.1
- man/new nologin.1
- man/new xbfd.1m
- man/new collectd.3
- man/new collectd.conf.4
- man/new collectd-*.4
- man/new types.db.4
- man/new tty_columns.3
- man/new select_columns.3
- man/new tty_variable.3
- man/new input_targets.3
- man/chg input: input_targets added to SEE ALSO section.
- man/new pl.1
- man/chg edrcsetup.1m: reference to patchinstall.1m.
- man/chg edrcsetup.1m: section REQUIREMENTS, additional references in section SEE ALSO, additional information to the ~edrc/bin/sys configuration.
- man/chg pack.1m: reference to edrcsetup.1m.
- man/cor pack.1m: formatting in SEE ALSO section.
- man/chg patchinstall.1m: reference to edrcsetup.1m.
- man/chg edrcrevision.1: action/object to object/action.
- man/chg edrcrevision.1: separate releases using subsections.
- man/new termcap.4
- man/chg manpages.4: additional escape sequences, simplification, new section title for used troff commands.
- man/cor sqlite.1: setting of quotations.
- man/new pmdesc.1
- man/new CGI.3
- man/new CGI::Auth.3
- man/new CGI::Cookie.3
- man/new DBD::DBM.3
- man/new DBD::SQLite.3
- man/new DBI.3

- man/new HTML::Parser.3
- man/new MIME::Lite.3
- man/new perl_modules.3: lists all provided perl module man pages.
- man/new perl_wrapper.1
- man/new ksh_wrapper.1
- man/new html2mht.3
- man/new perlenv.3
- man/fix logrotate.cfg.4: description of LOGBASE default.
- man/fix makeuser.cfg.4: description of LOG default.
- man/new is_osid.3
- man/cor select_columns.3: example 2.
- man/cor sysconfig.1: example 3 output.
- man/chg vsdfml: can receive text thru a pipe via stdin when using the pseudo file name -.
- man/chg fit2width.3: optional width option.
- man/chg regexintro.4: remove description of sub-expressions.
- man/new hostaliases.3
- man/chg checkopt.3: example 1 code.
- man/chg filedist.1: hostaliases(3) to SEE ALSO section.
- man/chg rcmd.1: hostaliases(3) to SEE ALSO section.
- man/chg log: add edrcintro(1) to SEE ALSO section.
- man/new fssum.1.
- man/chg compatibility: bundled perl modules are listed, too.
- cmd/fix sw_report: argument parsing of -h option.
- doc/new edrc_manpages-<VERSION>.zip: a zip file containing all documentation files is added to the edrc/doc directory and referenced by the EDRC.1 man page and the EDRC.html title page.
- cmd/chg tsize: performance increase.
- cmd/chg cmon: allow execution HP-1* -> HP-.*.
- cmd/chg cmswitch: allow execution HP-1* -> HP-.*.
- cmd/new Solaris/zdiff: new.
- cmd/new Solaris/zgrep: new.
- cmd/new Solaris/zless: new.
- cmd/new Solaris/zmore: new.
- cmd/chg edrcenv: more precise EDRC_DEBUG_* environment variable selection.
- cmd/chg cmon: improve compatibility, avoid "command not found" error on systems, where MC/ServiceGuard is not installed.
- cmd/new select_columns: filter to select named columns from stdin and print it to stdout.
- cmd/chg approot: avoid "permission denied" error when current working directory has specific permission settings especially on Solaris.
- cmd/chg thttpd: usage message extended.
- opt/chg thttpd: status to query the status of the webserver started with thttpd start.
- cmd/fix pscount: check if maximal process counts are numbers.
- cmd/chg apropos: fit output to terminal width.
- cmd/chg whatis: fit output to terminal width.
- cmd/new Solaris/sysinfo: compiled for Solaris.
- cmd/chg timezone: avoid error output on system wide timezone resolution on Solaris.
- cmd/chg timezone: enhance time zone resolution on Linux.
- cmd/chg cfg2html: version 1.6.4 for Solaris.
- cmd/chg cfg2html: version 4.23 for HP-11i and HP-11ia.
- cmd/chg cfg2html: version 1.58-2009/02/18 for Linux.
- cmd/chg csv2worksheet: use the new POI Java library poi-3.2-FINAL-20081019.jar.
- cmd/new is_existing: resolve if a file/dir/link exists without hanging if the resource (e.g. NFS) is not available.
- cmd/new Solaris/gunzip: as symlink to gzip.

- cmd/chg shell: do not set the USAGE_PATH environment variable.
- cmd/chg usage: include the directories edrc/lib/kshrc, edrc/lib/perl, edrc/lib/daemon to the usage output.
- cmd/new collectd: daemon to collect system statistics.
- cmd/new portscan: scan ports of a target system and return the ports that answer.
- cmd/chg is_up: when ping does not succeed to test if the remote host is up also try to connect using alternate network ports.
- cmd/chg shell: do not use the \$HOME environment variable to resolve the user's home directory.
- cmd/chg edrc: break multi line menu entries.
- cmd/chg tty_variable: performance increase by avoiding a tempfile.
- cmd/chg edrc: a forced line break can be added to a menu entry by entering an ";".
- cmd/chg shell: set the \$USER environment variable via the id command.
- cmd/chg Linux/lsof: new version 4.81.
- cmd/chg bin/*/lsof: use more compatible method to resolve newest installed binary version.
- cmd/chg bin/*/mc: use more compatible method to resolve newest installed binary version.
- cmd/chg bin/*/cfg2html: use more compatible method to resolve newest installed binary version.
- cmd/chg bin/*/nano: use more compatible method to resolve newest installed binary version.
- cmd/chg bin/*/bzip2: use more compatible method to resolve newest installed binary version.
- cmd/chg lib/*/expect: use more compatible method to resolve newest installed binary version.
- cmd/chg passwdsyncd_apply: ensure the entry of the command name and the correct session (=pid) into the log file passwdsyncd.log.
- cmd/chg ini.*: set \$HOME explicitly.
- cmd/chg usage: set \$HOME explicitly.
- cmd/new pl: ploticus.
- cmd/chg: thttpd can now distinguish between different web applications.
- cmd/chg bin/*/lsof: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg bin/*/mc: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg bin/*/cfg2html: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg bin/*/nano: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg bin/*/bzip2: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg lib/*/expect: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg bin/Linux/truss: react on EDRC_DEBUG_OS_WRAPPER, due to the fact that this script is "the longer arm" of the .os_wrapper command.
- cmd/chg edrc: ensure terminal initialization, if the configured terminal does not exist on the system.
- cmd/chg woist: ensure execution, even when csh is not available on the system.
- cmd/chg repeat: ensure execution, even when csh is not installed on the system.
- cmd/chg Linux/splitvt: start splitvt-<Version> and set the TERMCAP variable on systems that do not have /etc/termcap installed.
- cmd/chg edrc: ensure remote \$HOME resolution, even when csh or ksh is not installed on the remote system.
- cmd/chg edrc: enhanced localhost resolution when distributing scripts and establishing trunks.
- cmd/chg passwdsyncd: ensure remote \$HOME resolution, even when csh or ksh is not installed on the remote system.
- cmd/chg cronhandler: ensure execution, even when csh is not installed on the system.
- cmd/chg Linux/sqlite: version 3.6.13.
- cmd/new pmdesc: print all installed perl modules.
- cmd/chg .perl_wrapper: structure of bundled perl modules.
- cmd/fix .perl_wrapper: module (@INC) initialization.

- cmd/chg .perl_wrapper: set the LD_LIBRARY_PATH and SHLIB_PATH to the osid related libs directory as done in the os_wrapper.
- cmd/chg .os_wrapper: check if wrapped program file exists and return an appropriate error message otherwise.
- cmd/chg .ksh_wrapper: ensure that .os_wrapper cannot be called directly.
- cmd/chg .ksh_wrapper: check if wrapped program file exists and return an appropriate error message otherwise.
- cmd/chg .os_wrapper: ensure that .os_wrapper cannot be called directly.
- cmd/chg .java_wrapper: enhanced error message if wrapped program file does not exist.
- cmd/chg .java_wrapper: ensure that .java_wrapper cannot be called directly.
- cmd/chg .daemon_wrapper: check if wrapped program file exists and return an appropriate error message otherwise.
- cmd/chg .daemon_wrapper: ensure that .daemon_wrapper cannot be called directly.
- cmd/fix du_report: ensure that du_report appears in ERROR messages.
- cmd/new perlenv: to print the enhanced perl environment variables.
- cmd/new html2mht: convert HTML file to single web archive file (MHT).
- cmd/cor thttpd: Usage message.
- cmd/fix pscount: message output contains correct scriptname.
- cmd/chg sw_report: allow external setting of PRINT_FIT2WIDTH. Default is 'False'.
- cmd/chg fssum: command redesign for enhanced compatibility, speed and program structure.
- cmd/new hostaliases: resolve aliases for a host.
- cmd/chg msg: avoid awk error when usage is not correct.
- cmd/fix crfile: error of unary test operator on line 222.
- cmd/new sparse: filter to identify sparse files.
- cmd/new contrib/license: print WA2L/edrc license terms.
- cmd/chg vsdfml: can receive text thru a pipe via stdin when using the pseudo file name -.
- cmd/fix vsdfml: correct error message output.
- cmd/chg print_header: set timezone.
- cmd/chg cmmon: set timezone.
- cmd/chg mkuser: set timezone.
- cmd/chg pscount: set timezone.
- cmd/chg sav: set timezone.
- cmd/chg fit2width: the width option is optional now.
- cmd/chg sw_report: use xbfd internally to resolve the filesystems.
- cmd/chg apply2file apply2sw_inventory asup bwcreate crfile dbrep du_report duvi edrcpack esat filedist filelink fssum locations logrotate mail_file makeuser omniutil osup pack psup sat sav sw_report swvi symlink sys sysconfig user_info vadsf whereami woist xlog ypxfr_all fields2swvi filesize fmatch freespace hostlist is_config_byhand is_osid listtemp maketemp mkuser passwdsyncd_apply pscount pslist remote_copy remote_shell removetemp role_option server_environment timezone cltrash logcut passwdcombine rsat svi syscp sysdiff sysvi trash untrash cronhandler joblog jobwatch logcheckd passwdsyncd patchinstall watchdog: more strict option parsing.
- cmd/chg filedist: enhanced localhost resolution.
- cmd/chg rcmd: enhanced localhost resolution.
- cmd/chg rcmd: print "...(localhost)..." message when executing command on localhost.
- cmd/new WA2L::Util: perl module.
- cmd/chg log: avoid 'permission denied' error.
- cmd/fix xlog: error message output.
- cmd/fix mail_file: error message output.
- cmd/chg contrib/clonemenu: clone hint in menu title splited to two lines.
- cmd/chg bwcreate catcomp checkopt cltrash cmmon cmswitch collectd cronhandler du_report edrc edrcenv filedist filelink fmatch fssum hostlist ini.bash ini.csh ini.ksh ini.sh ipsort job joblog jobwatch listtemp llcomp locations logtail logview lscomp lscp lsmv maketemp makeuser name omniutil pack passwdcombine passwdsort passwdsyncd passwdsyncd_apply patchinstall

pkg_hostname pscount pstree rcmd removetemp resolve_targetlist revision role_option
 server_environment sw_report sysconfig thttpd trash uncgz utgz untrash usage user_info
 whoisin ypxfr_all: unset GZIP environment variable to ensure correct gzip functionality, set
 LC_ALL=C to ensure consistent sorting.

- cmd/fix sysconfig: date format of comment in sysconfig settings file.
- cmd/fix pack: only return a package if the generation was completely successful.
- cmd/new shlib: print shared library path variables.
- cmd/fix fmatch: "cannot shift" error on Solaris.

1.5.17

Fri Dec 12 14:00:11 MET 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.17 (Thu Dec 11 20:22:26 MET 2008)
- cfg/chg kshrc: new alias termize to set the COLUMNS and LINES environment variables.
- cfg/chg kshrc: set COLUMNS and LINES variables prior to the call of vi, vim and view.
- man/chg edrcintro.1: short description of tsize.
- cmd/fix pscount: if min/max is not a number, handle it as it is not defined.
- cmd/fix pscount: avoid awk error output while escaping the characters while producing the output in
 pscount -p.
- cmd/chg edrc: set COLUMNS and LINES prior to start the viewer and editor.
- cmd/new tsize: print terminal size to be used to initialize the COLUMNS and LINES environment variables.
- cmd/chg checkopt: used option resolving changed to enhance compatibility.

1.5.16

Tue Dec 9 17:48:18 MET 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.16 (Thu Nov 27 00:01:44 MET 2008 release)
- cfg/chg kshrc: check if less is installed on the system, else use more.
- cfg/chg kshrc: set username in prompt based on id.
- man/chg EDRC.1: add readline.3.
- man/chg edrcintro.1: readline short description.
- man/chg contrib.edrc.1m: short description for contrib/edrc/sane.
- man/chg logcheckd.1m: moved pattern files from edrc/lib/logcheckd/pattern/ to edrc/var/logcheckd/pattern/.
- man/chg logcheckd.pattern.4: moved pattern files from edrc/lib/logcheckd/pattern/ to
 edrc/var/logcheckd/pattern/.
- man/chg logcheckd.cfg.4: moved pattern files from edrc/lib/logcheckd/pattern/ to edrc/var/logcheckd/pattern/.
- opt/new pscount: -p to print a process list of a system.
- cmd/new readline: read long input lines from stdin.
- cmd/chg input: support long line input on Solaris, too.
- cmd/chg many: internal improvement in querying osid.
- cmd/chg edrc: prompted trunks can now be canceled.
- cmd/chg edrc: when a file is edited or viewed this is reflected in the window title now, too.
- cmd/chg contrib/edrc/sane: stty sane as contributed command.
- cmd/chg pscount: print a summary line at the end of the process table showing the totals of the rows MIN,
 MAX, CURRENT and STATE.
- cmd/chg pslist: command performance tuning.
- cmd/chg pslist: more strict option parsing.
- cmd/chg pslist: no longer print a line feed, when no process was selected.
- cmd/chg title: performance enhanced.
- cmd/chg logcheckd: moved pattern files from lib/logcheckd to var/logcheckd
- cmd/chg sysconfig: ensure proper message output.
- cmd/chg passwdsyncd: ensure proper message output.
- cmd/chg scriptextract: ensure proper output on systems having bugous tr installed.
- cmd/chg patchinstall: change umask for Solaris.

1.5.15

Sun Nov 23 12:52:24 MET 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.15 (Sat Nov 22 18:32:42 MET 2008)
- man/chg edrcintro.1: description of pscount.
- man/chg edrcintro.1: man section of pslist.
- cmd/chg pslist: moved from bin to lib.
- cmd/new pscount: command to count system processes and evaluate if the situation correlates to a defined number of required processes.
- cmd/chg contrib/edrc/clonemenu: compatibility improvement to be able to run on Solaris, too.
- cmd/chg contrib/edrc/clonemenu: messages adjusted.

1.5.14

Fri Nov 21 12:56:24 MET 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.14 (Tue Nov 18 22:25:46 MET 2008)
- cfg/chg kshrc: .. alias removed.
- cfg/chg kshrc: usage strings adjusted.
- man/chg shell.1: review.
- cmd/fix maketemp: use correct POSIX compatibility setting for HP-*.
- cmd/chg maketemp: print activity "Unknown" for temporary directories where not all information could be determined to judge tempdir activity.
- cmd/chg maketemp: compatibility improvement.
- cmd/fix print_list: avoid awk error "escaped \. treated as plain .".
- cmd/chg random: new debug method.
- cmd/chg trash,untrash,cltrash: new debug method.
- cmd/chg tolower: compatibility improvement.
- cmd/chg toupper: compatibility improvement.
- cmd/chg log: compatibility improvement.
- cmd/chg msg: compatibility improvement.
- cmd/chg passwdsyncd: compatibility improvement.
- cmd/chg passwdsyncd_apply: compatibility improvement.
- cmd/chg sysvi,sysdiff,syscp,svi: compatibility improvement.
- cmd/chg logcheckd: compatibility improvement.
- cmd/chg edrc: log edrc version index on startup and program quit.
- cmd/chg lib/<OSID>/many: error output to stderr.
- cmd/chg bin/<OSID>/many: error output to stderr.
- cmd/chg pkg_hostname: further tuning.

1.5.13

Fri Nov 14 01:48:59 MET 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.13 (Mon Sep 22 13:51:46 CEST 2008)
- cfg/new timezone.cfg: configuration file for timezone.
- cfg/chg kshrc: TZ is set from timezone.cfg
- cfg/chg kshrc: more and less use tabstops of 4 to be consistent with the common vi/vim settings in WA2L/edrc and use a more alias on systems where the tabstop cannot be set in more.
- cfg/chg rcmd.cfg: new SUMMARY setting to print a summary of the remote command invocations.
- cfg/new perl_wrapper.cfg: to configure .perl_wrapper.
- cfg/new ksh_wrapper.cfg: to configure .ksh_wrapper.
- dir/new var/sw: to store the packed software to or store WA2L/edrc patches if no other location is specified in the enterprise.
- man/cor tolower.3: formatting.
- man/cor toupper.3: formatting.
- man/cor fmatch.3: formatting.
- man/chg sysconfig.1: step 8) of system configuration.
- man/cor sysconfig.1: examples output and description.
- man/chg edrcintro.1: tomorrow, yesterday, is_weekend description.

- man/chg edrcintro.1: today description.
- man/chg syscp: names in examples aligned to other manpages.
- man/chg days: YYYY-MM-DD format.
- man/cor sysinfo: examples.
- man/new yesterday.3
- man/new tomorrow.3
- man/new today.3
- man/new untgz.1
- man/new timer.1
- man/new fit2width.3
- man/new pstree.3
- man/new indent.3
- man/new ident.3
- man/chg edrcintro: indent/ident descriptions.
- man/new group.3
- man/new is_weekend.3
- man/new sysvi.1
- man/new svi.1
- man/chg edrcintro: checkopt description.
- man/chg edrcintro: lib/perl/pm description.
- man/chg edrcintro: user description.
- man/chg checkopt.3
- man/new uncgz.3
- man/new user.3
- man/new sys.1
- man/chg edrcsetup.1m: reference to sys (1), use of edrc/bin/sys to start the WA2L/edrc maintenance menu.
- man/chg log: \$TZ environment variable.
- man/chg days: \$TZ environment variable.
- man/new pslst.1
- man/chg approot.3: SEE ALSO section.
- man/chg edrcroot.3: SEE ALSO section.
- man/chg rcmd.cfg.4: description of SUMMARY setting.
- man/new is_up.3
- man/chg EDRC.1: include hint about starting the thttpd to view documentation from remote.
- man/chg edrc.1m: add description to new trunk type prompt:nolocal for defining trunks with no Localhost selection.
- man/chg edrc.1m: add description to new edrc command tz to set the time zone within edrc.
- man/chg edrc.1m: description of the new \$EDRC_DEBUG environment variable.
- man/chg edrc.cfg.4: add tz to DENY_LIST description.
- man/new sqlite.1
- man/chg crfile.1: -a option (=accelerated file creation for root) removed.
- man/new shell.1
- man/chg listtemp.1: description of the PROCESS column output.
- man/chg server_environment.3: description of the -R option including an additional usage example.
- doc/new sqlite-3.6.3.html
- cmd/chg days: now accepts dates in the format YYYY-MM-DD, too.
- cmd/new today: return date of today.
- cmd/new yesterday: returns date of yesterday.
- cmd/new tomorrow: returns date of tomorrow.
- cmd/new is_weekend: returns True on weekend days (Saturday and Sunday), else False.
- cmd/chg cfg2html: version 1.63 for Solaris
- cmd/chg sat: more strict option parsing, ego message is no longer printed when -V or -s is used.

- cmd/new DocBrowser: display the documentation in WA2L/DocBrowser, too.
- cmd/fix fit2width: avoid error on systems where awk has restriction on number of fields.
- cmd/chg edrc: .sav directories are no longer distributed when invoking the distribute command in edrc.
- cmd/chg edrc: the dialog of the trunk command in edrc accepts now also the '.' to specify the current directory in addition to the '' input.
- cmd/chg edrc: the terminal window title is set to <user>@<hostname> when started. When trunks are established, the title will also represent the user and host where currently connected to.
- cmd/chg edrc: the timezone is set from the timezone.cfg file.
- cmd/chg edrc: print a INFO message when a newer edrc [index] is installed on the system.
- cmd/fix edrc: do not dive into submenu when trunk prompt on a prompted trunk is interrupted pressing Ctrl+C.
- cmd/chg edrc: new trunk type prompt:nolocal to provide a trunk target list excluding the Local: entry.
- cmd/chg edrc: TRUNK TARGETS numbering changed to provide an identical numbering for trunk types prompt and prompt:nolocal.
- cmd/chg edrc: avoid "No Type" error on terminal initialization.
- cmd/chg edrc: new command tz to set the timezone within edrc.
- man/chg edrc.1m: description of new trunk type direct:nolocal.
- cmd/fix contrib/scriptstats: correct output on missing diff dist file.
- cmd/new indent: identical to the already existing ident command, that is depreciated now. ident will continue to exist due to compatibility reasons.
- cmd/chg makeuser: use indent internally instead of ident.
- cmd/chg omniutil: use indent internally instead of ident.
- cmd/cor print_list: usage text spelling.
- cmd/chg print_list: if the \$PRINT_FIT2WIDTH is set to False the output is not limited to the terminal width.
- cmd/chg input: less aggressive tty restore on abort.
- cmd/new group: print the group name for a given group-id.
- cmd/new checkopt: check options.
- cmd/chg usage: internal improvements.
- cmd/chg name: use less on Linux.
- cmd/chg revision: use less on Linux, three digits for revision number.
- cmd/new title: set the terminal title.
- cmd/chg shell: /usr/ucb is included in the \$PATH now, too.
- cmd/chg shell: the terminal window title is set to <user>@<hostname> when started.
- cmd/chg server_environment: now caches the resolved information.
- opt/new server_environment: -R to print the whole record as variables.
- cmd/chg server_environment: performance improvement.
- cmd/new sqlite: sqlite version 3.6.3 .
- cmd/chg .perl_wrapper: now supports the use of additional Perl modules installed in edrc/lib/perl/pm .
- cmd/chg .perl_wrapper: can be configured in perl_wrapper.cfg, now.
- cmd/new uncgz: unpack gzipped cpio file(s).
- cmd/new user: return user name for a given uid.
- cmd/new sys: edrc short start to load the edrc.sys.cfg configuration that is provided for WA2L/edrc maintenance.
- cmd/new timezone: returns the timezone.
- cmd/chg many: set timezone (TZ) environment variable.
- cmd/chg mc: restore terminal window title after exit.
- cmd/chg mcview: restore terminal window title after exit.
- cmd/chg mcedit: restore terminal window title after exit.
- cmd/new wget: for Solaris.
- cmd/chg swvi: output speed up when resolving the primary group of the RUN_USER.
- cmd/chg du_report: exclude index files in parallel WA2L/edrc installations.
- cmd/chg du_report: use standard header.

- cmd/chg du_report: exit code=2 when a non-root user calls it.
- cmd/new pslist: list and select processes in a cross operating system compatible format.
- cmd/chg rcmd: if configured in the rcmd.cfg file a summary of succeeded/failed remote command invocations is printed.
- cmd/chg filedist: if configured in the filedist.cfg a summary of succeeded/failed file distributions is printed.
- cmd/chg uid: internal implementation changed to ensure a more consistent output.
- cmd/new tz: to set the \$TZ variable in the shell.
- cmd/new debug: to toggle the \$DEBUG variable in the shell.
- cmd/new edrcdebug: to toggle the \$EDRC_DEBUG variable in the shell.
- cmd/fix uniqpath: no error when -h is invoked.
- cmd/chg .java_wrapper: review, use uniqpath internally.
- cmd/chg .java_wrapper: internal improvement.
- cmd/chg .ksh_wrapper: internal improvement.
- cmd/chg .perl_wrapper: internal improvement.
- cmd/fix choice: do not exit, when no default value is specified and the input is empty.
- cmd/chg all: internal replace of \$DEBUG with \$EDRC_DEBUG and \$EDRC_DEBUG_<COMMAND-NAME> to separate command debugging completely from script debugging.
- cmd/chg shell: set \$EDRC_DEBUG to False on startup.
- cmd/chg shell: startup mode to ensure online patching.
- cmd/chg pkg_hostname: avoid error output when getcwd fails.
- cmd/new getmountpoint: compiled for Solaris.
- cmd/new getfilesystem: compiled for Solaris.
- cmd/fix crfile: now runs on Solaris also successful.
- cmd/chg crfile: -a option (=accelerated file creation for root) removed.
- cmd/chg edrcenv: displays defined EDRC_DEBUG_* variables, too.
- cmd/chg pkg_hostname: tuned.
- cmd/chg maketemp: new process resolution uses pslist, now. PPID and process resolution changed.
- cmd/chg maketemp: enhanced flagfile format.
- cmd/chg listtemp: displays information according to the new flagfile format. The status of temporary directories created with the old flagfile format are still displayed correctly.
- cmd/chg sysvi: export the editor settings also to vim.
- cmd/new print_header: print a standard report header.
- cmd/fix Solaris/lynx: prevent "cannot open terminal" error.
- cmd/new print_index: print a standard column index.
- cmd/chg sw_report: uses maketemp to handle tempfiles, now.
- cmd/chg sw_report: internal improvements.
- cmd/chg sw_report: exclude index files in parallel WA2L/edrc installations.
- cmd/chg scriptheadersync: uses maketemp to handle tempfiles, now.

1.5.12

Sun Sep 21 11:56:04 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.12 (Sat Sep 20 17:02:11 CEST 2008)
- cmd/chg makeuser: when querying uid, sort output list by UID field, when querying gid, sort output list by UID field.
- man/cor freespace: calculation example.
- man/chg filesize: calculation example.
- cmd/chg input: on Ctrl+C, input exits with exit code 2.
- man/chg input: Exit status 2.
- man/new fmatch.3
- man/fix edrcintro.1: replace \$DATA_FILE by \$FMATCH_DATA_FILE.
- man/chg regexintro.4: hint to avoid \{m,n\} construct for compatibility reasons.
- man/new tolower.3
- man/new toupper.3

- cmd/fix tolower: usage message.
- cmd/fix toupper: usage message.

1.5.11

Thu Sep 18 22:40:36 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.11 (Thu Sep 11 21:15:49 CEST 2008)
- cmd/fix edrc: on errors on scriptextract count errors as failure.
- cmd/chg scriptextract: exitcode > 0 on all errors.
- cmd/new freespace: print freespace of a directory on a filesystem.
- cmd/fix pack: computes relative directories specified in -d correctly, now.
- cmd/new filesize: print the size of a file.
- man/chg edrcintro.1: freespace and filesize description.
- man/chg sysconfig.1: EXAMPLES enhanced.
- man/chg sysconfig.1: description of the use of CONFIG_BYHAND setting.
- man/chg sysconfig.cfg.4: description of CONFIG_BYHAND setting.
- man/new freespace.3
- man/new filesize.3
- cmd/fix hostlist: cache computation on HPUX.
- cmd/chg sysconfig: avoid creation of .EDRC.config_byhand. and .EDRC.sysconfig. when the name file is empty.
- opt/new sysconfig.cfg: CONFIG_BYHAND setting to enable replacing (REPLACE) or merge (MERGE) of the applied config_byhand file entries. Default is REPLACE as the behaviour before the existence of the CONFIG_BYHAND option.

1.5.10

Sun Sep 7 14:03:11 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.10 (Mon Aug 18 12:14:23 CEST 2008)
- cmd/fix print_list: avoid syntax error on Solaris when calling "cat file | print_list = 2".
- cmd/chg patchinstall: ensure corpse remove counting on Solaris, too. "file does not exist" error suppressed when backing up files to be patched.
- cmd/chg sysconfig: change umask for Solaris operating systems to ensure proper target file permissions. The environment variable SYSCONFIG_ROOTDIR and SYSCONFIG_FORCE is also exported to the check script, now.
- cfg/chg kshrc: export EDITOR variable, to enable crontab editing.
- cmd/chg hostlist: now supports a resolution caching functionality to improve performance in complex hostlist.cfg definitions.
- cfg/chg hostlist.cfg: new settings CACHE and CACHE_DEPENDENCY to support the resolution cache.
- cmd/chg shar.setup shar.install/install: when installing the WA2L/edrc shell archive, free space in /tmp is checked now, too. The \$TMPDIR environment variable can be set to point to a directory having sufficient free space. The installation logfile in /tmp now has the process id of the installation in the name, too.
- cmd/fix shar.install/install: when only extracting the package, this is also possible now if the package name is not edrc. When installing on Solaris permissions were not set correctly when the umask of the calling shell was not set to 000, this workaround is now no longer needed.
- man/chg edrcsetup.1m: description of the \$TMPDIR variable and the checked space requirements in the temporary directory and the installation directory.
- cmd/fix crfile: tempfile cleanup.
- cmd/new uid: return the UID of a given user.
- cmd/new gid: return the GID of a given group.
- cmd/new gecost: return the gecost (=comment) field of the user account.
- cmd/fix edrc: ensure user name setting in script templates, _env and _trunk settings even when \$USER in the calling shell is not set.
- cmd/fix edrc: menu points were sometimes not displayed correctly on some Linux versions and on Solaris.
- cmd/chg edrc: list failed/aborted hosts at the end of distribute command completion.

- cmd/fix edrc: correct counting of success/fail/abort for distribution summary.
- cmd/chg vsdfml: no longer influence to the tty settings.
- cmd/chg input: no longer influence to the tty settings.
- man/new uid.3
- man/new gid.3
- man/new gecost.3
- cmd/chg makeuser: resolve information from various name services, list information from LDAP, too.
- man/chg makeuser.1: description of LDAP, NIS and FILES use as name service source.
- man/chg hostlist.3: add reference to edrc/var/samples/hostlist/ where configuration examples can be found.
- man/chg server_environment.3: add reference to edrc/var/samples/hostlist/ where configuration examples can be found.
- man/chg pack.1m: add example 0) where the most simplest usage of pack is shown.
- man/chg patchinstall.1m: new example to remove corpses without installing a new patch.
- man/chg sysconfig.1: more information in DESCRIPTION section.
- man/new ll.1
- man/new envpasswdstrip.1
- man/chg manpages.4: add COPYRIGHT section to manpage structure description.
- man/chg *.*: add COPYRIGHT section to all manpages of WA2L/edrc .

1.5.09

Sun Aug 17 15:34:51 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.09 (Thu Jul 24 18:25 CEST 2008 release)
- man/new mail_file.1
- man/new mail_file.cfg.4
- man/chg edrcintro.1: mail_file.cfg (4) added in SEE ALSO section.
- man/new makeuser.1
- cmd/chg makeuser: return exit code 2 when makeuser is not started with the correct user account
- man/new makeuser.cfg.4
- man/new userclass.index.4
- cmd/chg .perl_wrapper: add /opt/perl/bin to standard path search due to perl installations on HP-UX. use whereami instead of woist when perl was not found in standard path search.
- cmd/fix .perl_wrapper: ksh removed as execution shell when perl was not found in searched locations.
- man/new trash.cfg.1
- cmd/chg server_environment: internal improvement.
- cmd/fix shar.install/install: install log error messages when package could not be installed.
- man/cor filedist.1: orthography.
- man/new pkg_hostname.3
- cmd/fix omniutil: do not apply the template if the corresponding version directory does not exist in the template .tar.gz file.
- cmd/chg omniutil: use ident and do_exit() internally, exit codes aligned.
- cmd/chg input: ensure proper terminal behaviour when using the NO_ECHO option and the user presses Ctrl+C during input.
- man/new omniutil.1
- man/new omniutil.cfg.4
- man/new usage.1
- man/new maketemp.3
- cmd/new fmatch: return values selected for a certain key.
- cmd/fix remote_copy: failing (no connect) error when connection cache and NO_PROBE_INTERVAL indicated to do the connection probing.
- cmd/new ll: beside the ll alias in the shell, have the ll command also available as binary. This helps if rcmd is invoked on systems that do not know the ll command.
- man/new removetemp.3
- cmd/fix sysconfig/tools/HP-11/edit_netconf: tempdir cleanup.

- cmd/chg mc: remove locale files from package.
- cmd/chg nano: remove locale files from package.
- cmd/chg httpd: usage message and option parsing changed.
- cmd/chg edrcman: use uniqpath in \$PATH and \$MANPATH.
- dir/new var/samples/: configuration samples.
- man/new listtemp.3
- cmd/chg cfg2html: version 1.51 for Linux, version 3.66 for HP-.*.

1.5.08

Sun Jul 20 14:29:49 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.08 (Sat Jun 14 01:23:13 CEST 2008 release)
- cmd/chg hostlist: more efficient HOSTGRP resolution.
- cmd/chg server_environment: output order when calling server_environment -l.
- cmd/new shell: shell having a similar environment as the shell started thru the shell command within edrc.
- cmd/chg vsdfml: use xmore only when started within edrc, avoid terminal echo suppression when exiting vsdfml with Ctrl+C.
- man/new regexintro.4
- cmd/new uniqpath: removes duplicate entries from a colon separated string as known from the \$PATH, \$MANPATH, \$USAGE_PATH.
- cmd/chg untgz: code formatting.
- cmd/chg .os_wrapper .perl_wrapper .ksh_wrapper .daemon_wrapper sysconfig edrc watchdog rcmd filedist: remove duplicates in \$PATH setting using uniqpath.
- cfg/chg kshrc: use uniqpath in ap, cwdadd, addc wd.
- man/new uniqpath.3
- cmd/chg untgz: print usage message when no files are specified.
- cmd/new expect: programmed dialog with interactive programs.
- cmd/fix edrcman: Osid dependent \$MANPATH is now correctly resolved.
- cmd/chg nano: include ncurses libraries for Solaris.
- cmd/new connect: make socket connection using SOCKS4/5 and HTTP tunnel.
- man/chg logcheckd.1m: example added.
- man/chg logcheckd.pattern.4: example added.
- man/new libexpect.3
- cmd/chg lib/<OSID>/libs: move os dependent libraries to edrc/lib/<OSID>/libs to eliminate duplicates and to simplify upgrades.
- cmd/chg .os_wrapper: set LD_LIBRARY_PATH and SHLIB_PATH if the command is installed with WA2L/edrc and point to edrc/lib/<OSID>/libs.
- cmd/new bz* for Solaris
- man/new bz*.1 for Solaris
- cmd/chg lib/<OSID>/tcpdump: direct startup, because library settings could be moved to .os_wrapper.
- cmd/chg lib/<OSID>/rsync: direct startup, because library settings could be moved to .os_wrapper.
- cmd/chg lib/<OSID>/cfg2html: dynamic version resolution.
- cmd/chg lib/<OSID>/nano: no more use of tail -l.
- cmd/chg edrcman: the environment variable to be set to check other operating systems man pages is \$MAN_OSID and no longer \$OSID.
- man/chg edrcman.1: change \$OSID to \$MAN_OSID.
- cmd/fix logview, logtail: repeat selection output on wrong selections.
- cmd/fix crfile: error on free space resolution on systems having long special file path.
- cmd/new patchinstall: patch a WA2L/edrc installation.
- cfg/new patchinstall.cfg: configuration file for patchinstall.
- cmd/chg revision: filter selection to avoid syntax errors.
- cmd/chg name: filter selection to avoid syntax errors.
- cmd/new input_target: special purpose input to query lists of hosts.
- cmd/chg tty_variable: more strict variable resolution.
- cfg/chg kshrc: save \$TERM, tty erase.

- cmd/chg edrc: new edrc command 'top' to browse to the root menu.
- cmd/chg shar.install/install: save install logfile to edrc/var/log/install.log, too.
- man/chg edrcsetup.1m: description of edrc/var/log/install.log
- doc/chg edrc_manpages-<VERSION>.html/*.html: Package: WA2L/edrc <VERSION> is added to html page.
- cmd/chg logtail: first print the terminal size to screen.
- cmd/chg mkuser: speed up execution.
- cmd/chg logtail: resolution of tilde syntax (~user) in config file.
- man/new patchinstall.1m
- man/new patchinstall.cfg.3
- cmd/chg .os_wrapper: add Stdlib_path to LD_LIBRARY_PATH and SHLIB_PATH.
- cmd/chg ident: moved to lib/perl, use of .perl_wrapper for startup.
- cmd/chg lynx, lsof, du_report, crfile, filelink, revision, is_running, remote_copy, user_info, mc, script-stats, doc, logs, dirwatch: replace the usage of tail -<rows> with the more portable sed command.
- cmd/chg edrc: distinguish between the different tail -<rows> variants.
- cmd/chg pack: performance increase of pack -L by factor 14. This causes a performance increase by factor 5 of patchinstall.
- cmd/chg csv2worksheet: use the new POI Java library poi-3.1-FINAL-20080629.jar.
- cmd/chg catcomp: avoid error "/opt/edrc/bin/catcomp: -: does not exist".
- man/new logrotate.1
- cmd/chg logrotate: exit code on missing configfile changed from 1 to 3.
- cmd/fix vsdfml: printing of terminal control characters on HP-UX when using vsdfml from outside of edrc.
- man/new logrotate.cfg.4
- cmd/chg logrotate: now sets defaults when LOGBASE or DEFAULT_SAVES in config file are not set.
- cmd/chg logcut: use of library functions, -p output changed
- cmd/fix du_report: avoid error messages on missing index files when using the index file cache file.
- man/new logcut.cfg.4
- cmd/fix edrc: print correct path on established trunks.

1.5.07

Sat Jun 7 20:13:37 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.07 (Sun May 11 12:53:17 CEST 2008)
- cmd/fix name: fix error when using command in directories where a directory with the name *text* or *script* exists.
- cmd/fix revision: fix error when using command in directories where a directory with the name *text* or *script* exists.
- cmd/chg listtemp, maketemp, removetemp: convert relative pathnames given in -d option to absolute.
- cmd/chg logcheckd: hostname is added to logfile names, section markers added to HTML report output to allow report post processing.
- cmd/chg logcheckd: tooltips added to overview table and logfile section levels.
- man/chg vsdfml.3: magic key in example changed from /bin/ksh to /bin/sh.
- cmd/chg contrib/pstree: ensure that only process id numbers can be specified.
- cmd/chg edrc: trunk behaviour: session names are transferred to remote hosts, session names remain read only when source host session name was read only.
- cmd/chg edrc: it is now ensured that ad hoc session names are unique, too.
- cmd/chg edrc: when setting the session name with the -n option the @ID@ field provides the functionality to set the name of the session with an ID part that is either the process id of the started edrc session or, if a session with the same name already exists, a random value. This avoids session name clashes of session names set by hand in a started edrc session by the 'name' command and sessions started with the -n option.
- cmd/chg edrc: avoid the creation/renaming of menupoints and menus to command equivalents.
- cmd/chg sat: sat now starts edrc with -n SAT_@ID@ and no longer with -n SAT_\$\$ to support session name transfer thru trunks and to avoid session name clashes.

- man/chg edrc.1m: description of @ID@ field and behaviour of transferring session names to trunk targets.
- man/chg logcheckd.1m: description of report structure added.
- cmd/fix edrc: on unsuccessful trunk establishment due to 'host not up' no longer dive into the local sub-menu.
- man/chg sat.1 asup.1 psup.1 osup.1: description of <id> assignment adjusted to the new @ID@ feature for naming an edrc session.
- cmd/fix contrib/doc: suppress multiple empty lines in output when running on Solaris, too.
- cmd/chg edrc: correct initialization of vim on systems where vim is started thru the vi command.
- cmd/chg contrib/logs: correct initialization of vim on systems where vim is started thru the vi command.
- cmd/fix duvi: missing initialization of trap signals corrected.
- cmd/fix filedist: correct use of ASCII header option to allow proper cross operating system file distributions.
- cmd/fix shar.install/install: correct user creation on non-redhat Linux systems.
- cmd/chg shar.install/install: fail output if user/group creation generated an non-successful return value.
- cmd/chg osid: on duplicate osid.dat entries only the first matching entry is returned and not all entries.
- cmd/chg pack: -n option (=no timestamp update) replaced with -t (=timestamp update), no timestamp update is default, now.
- cmd/chg pack: more strict usage parsing.
- man/chg pack.1m: option -n replaced with -t, usage description more precise.
- cmd/new var/sysconfig/HP-11/edit_netconf: interactively edit /etc/rc.config.d/netconf formatted files on HP UNIX for usage in post_exec scripts in sysconfig.
- cmd/chg Linux/cfg2html: new version 1.45.
- cmd/chg HP-UX/lsof: new version 4.80.
- chg/src lsof: new version 4.80.
- cmd/chg HP-UX/nano: distribute libncurses.sl and libintl.sl also with WA2L/edrc
- cmd/chg mc: set TERM to 'xterm' if \$MCTERM is not defined.
- man/chg edrcintro.1: description of \$MCTERM added to the short description of mc.
- cmd/chg pack: application timestamp prefix can be specified separately from the APPLICATION_BASEDIR.
- cfg/chg pack.cfg: new option TIMESTAMP_PREFIX to separate the timestamp from the application prefix.
- cmd/chg pack: -l option also lists the timestamps for the different releases available.
- man/chg pack.1m: -l option description.
- man/chg pack.cfg.4: description of new option TIMESTAMP_PREFIX added.
- cmd/chg usage: -h message enhanced, internal improvements, USAGE_PATH environment variable recognized.
- cmd/chg makeuser: classes/.index file format with field COMPANY expanded, output format when entering '?' changed (now print_list is used internally).
- cmd/chg Linux/lsof: new version 4.80.
- cfg/new HP-11ia: directories prepared in edrc/bin, edrc/lib and edrc/man for Itanium version of HP-UX.
- cmd/chg homedir: answer speed increased by factor 200.
- cmd/chg woist: cleanup of internal function homedir().
- man/chg edrcman.1: sections ENVIRONMENT, EXIT STATUS and EXAMPLES added.
- cmd/chg maketemp: usage message enhanced with EXAMPLES.
- cmd/chg removetemp: usage message enhanced with EXAMPLES.
- cmd/chg listtemp: usage message enhanced with EXAMPLES.
- cmd/fix shar.install/install: package extract when package name does not start with 'edrc-'.
- cmd/new var/sysconfig/default/create_users: create users based on a passwd format input.
- cmd/fix sysconfig: \$EDRC_SCRIPTNAME is set correctly to the check script while executing it. Therefore if the msg command is used within the check script, the output is complete, now.
- man/chg csv2worksheet.3: reference of file edrc/var/csv2worksheet/io.xls in the FILES section.
- cmd/chg whereami: output of 'Customer:' added.
- cmd/chg edrc: when executing the 'distribute' command, specified hostgroups (@GRP) are resolved now, too.

- man/chg edrc.1m: description that @HOSTGRP can be entered in the target system specification during script distribution.
- cmd/chg contrib/calc: to exit the calculator the only command is 'exit', 'quit' is no longer accepted to be in line with the general edrc commands. Improved error handling, basic fraud handling.
- cmd/fix edrc: when running on Linux the viewing of logfiles could not properly handle compressed logfiles.
- cmd/fix contrib/logs: when running on Linux the viewing of logfiles could not properly handle compressed logfiles.
- man/fix logcheckd.1m: pattern file location corrected to lib/logcheckd/pattern.
- man/chg logcheckd.cfg: description of variables which are allowed to be used in the eval resolve_mode.
- cmd/chg is_osid: now the * (multiple characters) and ? (one single character) placeholders are supported.
- cmd/chg contrib/omnimon: support a more wide range of osids.
- cmd/chg contrib/drvstat: support a more wide range of osids.
- cmd/chg contrib/cmviewcl: support a more wide range of osids.
- cmd/chg cmswitch: support a more wide range of osids.
- cmd/chg cmmon: support a more wide range of osids.
- cmd/fix logcheckd: export \$TODAY variable to eval resolve_mode logfile specifications.
- cmd/chg print_list: now a left indent can be specified as an optional option. Therefore the construct 'cat file | print_list ":" | indent 2 | fit2width -m 'tty_columns'' can be replaced with 'cat file | print_list ":" 2', now.
- cmd/chg contrib/scriptstats: use the new indent of print_list.
- cmd/chg makeuser: use the new indent option of print_list.
- man/new sysconfig.tools.3
- man/chg sysconfig.1: reference to sysconfig.tools.3.
- cmd/new tolower: translate a given string to lowercase.
- cmd/new toupper: translate a given string to uppercase.
- cmd/new filegrep: grep for something in all underlying ASCII files, files in .sav directories and files saved with the sav command are excluded from the search.
- cmd/chg makeuser: output formatting.
- cmd/chg mkuser: transfer symbolic links to homedirectories, too.
- cmd/chg contrib/doc: permission change due to display problems on certain Linux operating systems.
- cmd/fix apply2sw_inventory: PATH setting corrected to avoid command hang on certain systems.
- cmd/chg logcheckd: more strict config file parsing.
- cmd/new HP-11ia: compiled commands for osid HP-11ia added: bunzip2, bzip2, bzcat, bzdiff, bzgrep, bzip2, cpio, gnutar, gunzip, gzip, lsof, lynx, mc, mcedit, mcview, nano, pwcrypt, rsync, splitvt, sysinfo, tcpdump, timer, truss, ttyplay, ttyrec, ttytime, tusc, unzip, wget, zcat, zdiff, zgrep.
- cmd/chg cltrash logtail logview rsat svi syscp sysdiff sysvi trash untrash: moved to edrc/lib/ksh, due to the fact that used statements are on Solaris only available in ksh.
- cmd/new untgz: unpack gzipped tar file(s).
- cmd/chg all: setting of \$PATH unified in all commands.
- chg/osid: can now distinguish between different machine types (hardware), too.
- man/new osid.cfg.4
- man/chg osid.dat: description of the new two osid.dat file formats.
- cmd/chg pack: add PACK_OSID_DAT_SFI and PACK_OSID_DAT_WA2L to sadm file.
- cmd/chg shar.install/install: consider osid.cfg and the two possible osid.dat formats.

1.5.06

Sun May 4 18:22:09 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.06 (Thu May 1 17:31:50 CEST 2008)
- man/new logcheckd.pattern.4
- cmd/fix logcheckd: no hyperlink for section [EXCLUDE] in report section 'Overview'.
- cmd/chg logcheckd: less aggressive report removal when using -a reset.
- opt/new logcheckd: -a verify, to verify a patternfile.
- man/chg logcheckd.1m: description of -a verify added.

- man/new logcheckd.style.4
- cmd/chg csv2worksheet: use the new POI Java library poi-3.0.2-FINAL-20080204.jar .

1.5.05

Fri Apr 25 21:56:58 CEST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.05 (Wed Apr 16 13:25:00 CEST 2008)
- cmd/chg shar.install/install: change uid/gid of connection dir edrc/var/connection/<user>/*
- cmd/fix pack: "bad substitution error" fixed. This happened when running pack on Solaris
- cmd/new logcheckd: a daemon to check logfiles for patterns and to create a consolidated findings report.
- cfg/new logcheckd.cfg: config file for logcheckd.
- man/chg edrcintro.1: logcheckd.1m added, files section completed.
- cfg/chg WA2L/edrc: var/ezreport, lib/ezreport, var/sfi directories removed
- cfg/chg usage.list: usage strings of commands in lib/daemon added
- man/chg edrcsetup: change the command of security file generation to ensure more straight forward setup.
- man/new logcheckd.cfg.4

1.5.04

Wed Apr 9 17:12:38 METDST 2008 release

- cfg/chg edrcrevision.cfg: new release 1.5.04 (Sat Feb 2 15:41:50 CET 2008)
- man/chg edrcsetup.1m: add shar execution output example.
- cmd/chg shar.install/install: avoid creating the homedir and do not create the secure group in Linux when user creation is selected.
- man/chg edrc.1m: example for exceptional advanced trunk specification
- cmd/chg edrc: the TARGET_MENUPATH setting in trunk specifications can now hold target menus only consisting of the path as displayed on the edrc prompt (e.g. /Support/ApplicationSupport) without having to specify the menupoint (e.g. /sup:Support/asup:ApplicationSupport). This enables the user to change the target menupoints and the trunk is still working.
- cmd/fix edrc: on advanced trunk specifications (using TARGET_MENUPATH), the path displayed in the prompt can be misleading, this has been fixed. If you establish advanced trunks from version 1.5.04 to 1.5.03 or vice versa the trunk will work properly, but the prompt won't not be correct.
- cmd/chg edrc: to print spaces in a trunk comment output (TYPE=prompt), they have to be replaced by a % in the definition.
- cmd/fix print_list: it is now possible to start print_list from outside of edrc -> shell, too.
- man/new filedist.1
- man/new filedist.cfg.4
- man/new filedist.block.4
- man/new rcmd.1
- man/new rcmd.cfg.4
- cmd/chg cmmon: move cmmon from edrc/contrib/edrc to edrc/bin to have the command available in the shell, too. cmmon is still available as contributed command.
- cmd/chg cmswitch: move cmswitch from edrc/contrib/edrc to edrc/bin to have the command available in the shell, too. cmswitch is still available as contributed command.
- man/chg edrcintro.1: added cmmon and cmswitch.
- opt/new server_environment.1: new option -C to print the customer of the environment.
- cmd/fix makeuser: check input of GID, UID, USERCLASS to ensure correct inputs.
- opt/new input: MAXLEN=len, MINLEN=len to check the input length.
- man/chg input.3: description of MAXLEN and MINLEN added.
- cmd/new sysdiff: list if a file applied with sysconfig differs to the file in the sysconfig repository.
- man/new user_info.1
- man/new user_info.cfg.4
- man/chg sysconfig: examples added.
- cmd/chg sysconfig: set EDRC_SCRIPTNAME to 'pre_exec'/'post_exec' when calling the pre- / post-exec scripts. From now on when using the msg command within pre- and post-exec scripts the correct prefix is printed.
- man/new is_sysconfig_byhand.3

- man/new hostlist.3
- man/new hostlist.cfg.4
- man/new server_environment.3
- man/new server_environment.cfg.4
- cmd/chg server_environment: remove unwanted characters from output
- man/new syscp.1
- opt/new sysdiff: -v for verbose output
- cmd/fix swvi: depreciated tempfile removal
- cmd/fix job, jobwatch, joblog: depreciated tempfile removal
- cmd/fix du_report: depreciated tempfile removal
- cmd/fix rsat: unused variable \$Tmp removed
- cmd/new .daemon_wrapper: start commands that act as daemons out of the lib/daemon/
- cmd/chg job, joblog, jobwatch, passwdsyncd, watchdog, cronhandler: start via .daemon_wrapper. From now on it is possible to patch those commands even if there is already an instance running.
- cmd/fix job, joblog, jobwatch: tempfiles are cleaned up properly now.
- opt/chg watchdog: -L option can now also list stop commands for the running watchdogs.
- cmd/chg pack: add the timestamp file of the release also to a PATCH.
- cmd/chg cronhandler: is now able to run on Linux and Solaris. cronhandler does now longer access the spool directory (except when running on Solaris the atjobs spool directory is accessed to get the content of a job), on ll other cases it is based now completely on the official cron/at commands.
- cmd/chg job: when entering a job the prompt is now "job>" instead of only ">".
- cmd/chg cronhandler: at queues are now also supported
- cmd/chg watchdog: temporary directory is /var/tmp and not /tmp any more.
- cmd/chg job: temporary directory is /var/tmp and not /tmp any more.
- man/chg passwdsyncd.1m: added exit code 11 description.
- cmd/fix fit2width: bug when running on Solaris fixed.
- cmd/fix server_environment: bug when running on Solaris fixed.

1.5.03

Sun Dec 16 14:04:55 CET 2007 release

- cfg/chg edrcrevision.cfg: new release 1.5.03 (Sun Dec 16 14:04:08 CET 2007)
- man/fix remote_shell: CONNECTION_MODE resolution
- man/fix remote_copy: CONNECTION_MODE resolution

1.5.02

Sun Dec 16 13:59:11 CET 2007 release

- cmd/chg edrc: edrcperm format output, location of lib/edrc.<name> files moved to lib/edrc/<name>
- chg/src Makefile: standardized, each command has an own src directory now
- cmd/chg lscomp, llcomp, catcomp: supports now .jar and .bw files, too
- man/chg lscomp.1, llcomp.1, catcomp.1: .jar and .bw added
- cmd/chg sysconfig: scripts located in var/sysconfig/tools/<OSID> and var/sysconfig/tools/default can be called from check-, pre_exec- and post_exec scripts.
- man/chg sysconfig.1: description of tool directories.
- cmd/new getmountpoint: print mountpoint of current working directory. This command is much faster as "bdf ." especially on HP-UX having large file systems.
- cmd/chg pkg_hostname: use getmountpoint if it exists for a certain osid otherwise still the not completely reliable method via the mount command is used as fallback.
- cmd/chg stat: can now return file permission as octal number.
- cmd/chg edrc: is no longer dependent on locally created 'edrc' user, the user can be created in any name service (NIS, NIS+, ...)
- cmd/chg passwdsyncd: is no longer dependent on locally created 'edrc' user, the user can be created in any name service (NIS, NIS+, ...)
- cmd/new lynx: a character oriented Web browser
- cmd/chg .os_wrapper: \$PATH is enhanced to edrc/bin and edrc/lib.

- cmd/new wget: the non-interactive network downloader
- cmd/new getfilesystem: print mountpoint of current working directory. This command is much faster as "bdf ." especially on HP-UX having large file systems.
- opt/new user_info: -c to specify a config file
- man/new getfilesystem.3:
- man/new getmountpoint.3:
- man/new days.3:
- man/new seconds.3:
- cmd/chg input: enhanced stability on computed input values
- opt/new input: IPV4, HH:MM, HH:MM:SS, NO_ECHO , MM/DD, MM/DD/YYYY
- opt/new remote_shell: -t to create a tty on remote system
- opt/new remote_shell, remote_copy: -f to force connection mode without connection cache involvement
- cmd/chg edrc: trunk mode support
- man/chg edrc.1m: description of trunk, renamescript, renamemenu command, guideline for differential, full distribution use
- cmd/new contrib/ping: send ICMP ECHO_REQUEST packets to network hosts as contributed command
- cmd/chg rsat: connection mode fix set to OpenSSH, due to the need of a TTY on a remote system
- cfg/chg rsat.cfg: CONNECTION_MODE setting removed
- cmd/chg filedist: the file to be distributed is no longer saved to /var prior to distribution. Messages on failure are more precise. Logging enhanced.
- cmd/chg remote_copy: show correct default connection mode in usage message
- man/new bwcreate.1m, bwcreate.cfg.4
- opt/new pack: -L to list files that would be packed
- man/new apply2sw_inventory.1
- cmd/chg apply2sw_inventory: exit status 2 returned if command was not able to write to the output directory is changed to 3 due to the usage of this exit code for two return situations.
- man/chg cfg2html.8: new cfg2html.1m
- man/new contrib.1m
- cmd/chg edrc: direct trunk invocation enhancement
- man/new input.3
- man/cor watchdog.1
- cmd/new print_list: format a csv list to a formatted list with dynamic column widths and adjustment to the current terminal width.
- man/new pack.1m, pack.cfg.4, edrcpack.1m
- man/chg several: use of .VB, VE makros to switch to courier font and back (seen in PDF file)
- cmd/chg remote_shell, remote_copy: cache handling changed to gain connection performance especially for forced connections.
- opt/new remote_shell.cfg, remote_copy.cfg: OPENSSH_OPTIONS to pass additional options to ssh and scp to influence the general behavior
- cmd/chg edrcman: react on external setting of \$OSID (Osid='osid' -> Osid=\${OSID:='osid'}).
- opt/new rcmd: -r [num] to retry rcmd calls on unsuccessful connections
- cmd/chg rcmd: has now a logfile, review
- cmd/chg edrc: command renamescript to rename an existing script, renamemenu to rename an existing menu
- cmd/new ttyrec: a tty recorder
- cmd/new ttyplay: play files recorded by ttyrec
- cmd/new ttytime: show times in files recorded by ttyrec
- man/new edrcrevision.cfg.4
- cmd/fix edrc: display of menupoint title if there is only one menupoint (=script) in a submenu
- man/new edrcperm.no_shell.3, edrcenv
- cmd/new maketemp: create a unique temporary directory
- cmd/new removetemp: remove a temporary directory created by maketemp
- cmd/chg edrc: tempfile handling enhanced

- cmd/chg many: tempfile changed to use maketemp and removetemp
- cmd/chg csv2worksheet: use new PIO version poi-3.0-rc4-20070503.jar
- cmd/new contributed/locations: print corporate locations
- man/new contrib.edrc.1m: list of contributed commands distributed with WA2L/edrc
- cmd/chg edrcman: add 1m to MANSECT. Therefore the .1m symlinks in the man stricture are not needed any more
- cmd/chg user_info: new field LOCKOUT. This field is set to y if any lock is set for a user account.
- cmd/chg contributed/doc: now documentation for menus can be displayed, too
- cmd/new mc: Midnight Commander
- cmd/new mcview: Midnight Commander file viewer
- cmd/new mcedit: Midnight Commander file editor
- man/new contrib.doc.1m
- cmd/new nano, rnano: an editor that can be restricted
- cmd/new nano.1, rnano.1, nanorc.4: manpages for nano and rnano
- cmd/fix edrc: menu point output corrected if menu point is not executable
- opt/new input: YYYY-MM-DD, the military date format
- cmd/chg input: leap years are checked, now
- cmd/chg contributed/cmswitch: enhanced for WA2L/edrc external use
- cmd/chg xmore: --MORE-- output goes to /dev/tty and not to stdout any more
- cmd/chg remote_shell: enable TTY allocation for rtools connections
- cfg/chg edrc.cfg: many more fine grained denials
- cmd/chg edrc: export edrc config file to recovery scripts and contributed commands in the \$EDRC_CONFIGFILE environment variable
- cmd/chg edrcenv: output of \$EDRC_CONFIGFILE environment variable
- cmd/new is_permitted: check if a functionality is permitted to be executed based on the DENY_LIST in edrc.cfg
- cmd/new contributed/logs: enhanced log viewer
- man/new vsdfml.3: very simple document formatting language
- cmd/chg edrc: help of contributed commands is resolved from contributed command header and no longer from *.help files
- cmd/new contributed/clonemenu: clone an existing menu
- opt/new choice: NOT_NULL, to prevent no input
- opt/new crfile: -a for accelerated usage (in some circumstances)
- cmd/chg crfile: allow usage for non-root users, too. In non-accelerated mode create a file consisting of zeroes. Space check corrected.
- cmd/chg shar.setup: obsolescent tail option +N replaced with -n
- cmd/chg shar.install.tar.gz/install: obsolescent tail option +N replaced with -n
- cmd/chg shar.install.tar.gz/install: enhanced post installation instruction output
- cmd/chg edrcsetup.1m: description of existing configurations expanded, ssh setup simplified
- cmd/chg cfg2html: actual versions cfg2html_hpux-b.3.46 and cfg2html_linux-1.40 integrated
- man/new: sat.1 asup.1 osup.1 psup.1
- cmd/chg edrc: avoid breakouts during trunk establishment
- cmd/new zdiff zgrep zless zmore
- man/new zdiff.1 zgrep.1 zless.1 zmore.1
- cmd/chg gzip: version 1.3.12
- cmd/chg csv2worksheet: using newest POI API poi-3.0.1-FINAL-20070705.jar
- cmd/chg edrc: use less as pager for Linux
- cfg/chg edrc.cfg: new option LOGCOMPRESS
- cmd/chg contributed/logs: is capable to handle compressed logfiles transparently
- cmd/chg edrc: is capable to handle compressed logfiles transparently, logheader of <contrib_cmd>.nolog is logged too
- man/chg edrc.cfg.4: option LOGCOMPRESS added
- cmd/chg edrcman: proper functionality even when no system wide man page configuration exists

- cmd/chg logtail: resolve method of user \$HOME
- cmd/chg logview: resolve method of user \$HOME
- cmd/new thttpd: tiny HTTPd
- cmd/chg cronhandler: review due to lost of a cron file after a system crash
- cfg/new cronhandler.cfg: new
- cmd/new listtemp: list temporary directories created with maketemp including state information.
- cmd/chg rcmd: logfile location can be configured in the config file now
- cfg/chg rcmd.cfg: new option LOG to define the log directory
- man/new cronhandler.cfg
- cmd/chg pack: improve timestamp preserving
- man/chg edrcsetup.1m: manual install instructions removed
- cmd/chg filedist: logfile location can be configured in the config file now
- opt/new user_info: -R prints complete user record in a processable form
- cmd/chg user_info: internal cache
- opt/new input: European date format DD.MM.YYYY, DD.MM
- man/cor edrcsetup.1m: formatting of user input commands
- man/new crfile.1
- man/new du_report.1
- cmd/new edrc: trunkbypass to globally bypass an active trunk
- man/new scriptheadersync.1
- cmd/chg install: check needed install space
- cmd/chg edrc: prevent compression of edrc.log when LOGCOMPRESS is set to True
- cmd/chg contributed/logs: add an additional line during output
- cfg/chg edrc.cfg: new option TMP_DIR
- man/chg edrc.cfg.4: new option TMP_DIR
- cmd/new contributed/scriptstats: print statistics about the recovery script menu tree
- man/new msg.3
- man/new log.3
- cmd/chg log: output to stderr if \$EDRC_LOGFILE is not set, set session to process id of calling script if \$EDRC_SESSION is not set
- cmd/chg edrc: enhancements of trunk options (TARGET_MENUPATH, TARGET_CONFIG) allowing hierarchical trunk configurations, performance tuning in trunk specification resolution

1.5.01

Sun Feb 4 23:15:04 CET 2007 release

- cmd/fix sav: save file in /etc/rc.config.d into a subdirectory even if the current working directory is not /etc/rc.config.d .
- opt/new hostlist: -l, to return the hostlist comma separated. Default is still to return a space separated list.
- man/new sav: source is SFI-QIF
- cmd/new resolve_targetlist: resolve a list of targets which may consist of hosts and hostgroups.
- cmd/chg user_info: output of encrypted password when calling user_info -u username
- cmd/new mail_file: send a mail with an attachment.
- cfg/new rcmd.cfg: SET_UMASK to set the umask on the remote system as it is on the local system.
- opt/new role_option: -c config_file to be able to specify a different configuration file.
- man/new edrcsetup: text moved from doc/install-steps.
- cmd/chg contributed/doc: structure and tags changed, speed increased
- cmd/chg edrcpack: default is to create a shell archive now.
- cmd/chg edrcpack: do not pack dot-files and directories from source directory
- cmd/chg input: log input to \$EDRC_LOGFILE, new options NO_LOG and LOG_STARS to control logging functionality and to suppress password logging to a logfile.
- cmd/chg change: log input to \$EDRC_LOGFILE, new options NO_LOG and LOG_STARS to control logging functionality and to suppress password logging to a logfile.
- cmd/chg user_info: LAST_LOGIN field returns the number of days since the last successful login.
- cmd/chg user_info: LAST_SMBACCESS field returns the number of days since the last samba access

- cfg/new user_info.cfg: new option SMBACCESSLOG
- cmd/fix edrcpack: log output corrected
- cmd/chg install: non verbose installation does log extraction to logfile
- man/new watchdog.1
- man/new watchdog.cfg.4
- man/new choice.3
- man/new name.1
- man/new apply2file.1
- man/new duvi.1
- man/new du_index.1
- cmd/chg sysconfig: check file- and directory permissions to install the files & directories, too
- man/chg sysconfig.1: file install description enhances with hint that file permissions and ownership are considered as well
- cmd/chg edrc: use remote_copy, remote_shell, log, msg commands instead of internal functions. logview alias removed. contributed commands lview and ltail renamed to logview and logtail. Output of contributed commands is logged depending of the existence of a <contrib_cmd>.nolog file in the contrib/edrc/ directory.
- cmd/new remote_copy: wrapper for remote copies (rcp, scp)
- cmd/new remote_shell: wrapper for remote shells (rsh, remsh, ssh)
- cmd/chg edrcpack: do not exclude ~edrc/.ssh directory from package, exclude connection/cache/*
- cfg/new remote_shell.cfg: config file for remote_shell
- cfg/new remote_copy.cfg: config file for remote_copy
- man/chg EDRC.1
- man/chg edrcintro.1
- man/chg edrc.1m
- man/chg edrc.cfg.4
- cmd/chg passwdsyncd: use remote_copy and remote_shell for synchronization
- cmd/chg filedist: use remote_copy and remote_shell for distribution. root is no longer needed to be allowed to connect from remote.
- cfg/new filedist.cfg: REMOTE_CMD_PATH to the cpio and gzip commands on remote systems.
- man/chg edrcsetup.1m: revision, enhancements
- opt/chg sat: -h displays no longer usage of edrc
- cmd/new rsat: remote start of sat
- opt/new tty_variable: [device], to specify the tty device if needed
- cmd/chg tty_variable: fallback to /dev/tty if the current terminal is "not a tty"
- cmd/chg filedist: log output enhanced
- cmd/fix filedist: input not logged to logfile any more
- cmd/chg rsat: put path to remote systems
- cfg/chg rsat.cfg: REMOTE_PATH to set the path on remote systems
- cmd/chg edrcpack: splitting of shell archives and patch archives, provide a setup script for splitted archives, complete cleanup of -a option
- opt/chg edrcpack: -g option removed
- cmd/chg edrc: menu display is now compatible with Suse- and FC4 Linux, parallel script distribution possible
- man/chg edrc.cfg.4: DIST_TYPE
- man/chg edrcsetup.1m: complete review, secure shell config description added
- cmd/chg apply2file, dbrep, du_report, logcut, passwdcombine, svi, sw_report, syscp, svi, scritpextract, sysconfig: *Linux* ECHO="echo -e" -> ECHO="/bin/echo -e"
- cfg/chg kshrc: shd function to show the \$DISPLAY variable content
- cmd/new ksh_wrapper: wrap ksh to avoid the #!/bin/ksh magic key
- cmd/chg syscp, sysvi, sw_report, du_report, apply2file, dbrep: #!/bin/ksh -> #!/bin/sh
- cmd/chg passwdcombine, logcut, random: starting via ksh_wrapper
- cmd/chg edrc: can be started without ksh, too

- cmd/fix is_config_byhand: evaluation of .EDRC.config_byhand files
- cmd/new pstree: print a process tree
- cmd/new dig: send domain name query packets to name servers
- cmd/chg install: cancel option added
- man/new mkuser.3:
- cmd/new apropos: search whatis database
- cmd/new whatis: search whatis database
- man/new apropos.1: search whatis database
- man/new whatis.1: search whatis database
- man/new passwdsyncd.1m: password sync daemon
- man/new lscp.1:
- man/new lsmv.1:
- doc/chg all: Makefile improved to generate the PDF, HTML, PS documents
- man/new trash.1
- cmd/new rsync
- cfg/new rsyncd.conf
- man/new rsync.1 rsyncd.conf.4
- cmd/new java: start java
- cmd/new java_wrapper: wrap java programs
- cmd/new bget: get HTTP
- man/new bget.1
- cmd/new seconds: calculate seconds since the Epoch for a given date and time
- cmd/new ident: continuous output with ident
- man/new manpages: manpages documentation
- cmd/new repeat: repeat a command multiple times
- man/chg edrcrevision.1: reformatted
- cmd/chg ini.ksh: review, ported to Linux
- cmd/fix watchdog: output channel assignment
- cmd/fix watchdog: environment variable setting to ensure nested watchdog starts
- opt/new watchdog: -L, to list all running commands as command line arguments
- cmd/fix passwdsyncd: output channel asininity
- man/new cltrash.1:
- man/chg trash.1: FILES section precised
- cmd/cor cltrash: file header
- cmd/fix remote_copy: connection cache read while probe the connection
- cmd/new csv2worksheet: load CSV data into an existing Excel spreadsheet
- man/new csv2worksheet.3:
- cmd/chg java_wrapper: method of loading the *.javaopt files changed to provide more flexibility
- cmd/new ccrypt: encrypt and decrypt files and streams
- cmd/new crypt: encrypt and decrypt files and streams using the old Unix crypt method.
- cmd/chg os_wrapper: checks if OS command is executable and if the filename exactly matches
- cmd/new sysinfo: print information on overall system statistics (uptime, ram, swap,...)
- man/new sysinfo.1:
- man/new lsof.1:
- cmd/chg filedist: more possibilities in controlling the blockage of file distribution
- cfg/chg filedist.block: format changed to 3 fields. To migrate current filedist.block file to the new format, add "block:" in the first row of each rule line.
- cmd/chg passwdsyncd: different encryption modes supported to enhance security
- cmd/chg passwdsyncd_apply: different encryption modes supported to enhance security
- cfg/chg passwdsyncd.cfg: ENCRYPTION_MODE (Enigma, AES, NO) to configure password information transfer
- man/new passwdsyncd.cfg:
- cmd/new splitvt: split a screen into two windows

- man/new splitvt.1:
- cmd/new xmore: more for use in recovery scripts
- cmd/new vsdfml: very simple document formatting language to be used in information "recovery scripts"
- cmd/new perl_wrapper: wrap perl scripts to ensure better portability.
- cmd/chg edrcpack: writes package information to var/edrcpack/sadm/ to provide more information about the package. This sadm file is part of patches and full releases.
- cmd/new zcat: symbolic links to gzip, where installed with WA2L/edrc
- cmd/chg cronhandler: crontab is loaded using the official crontab command, now
- cmd/chg edrc: header in menus, permission specification
- cfg/chg edrc.cfg: DENY_LIST
- opt/new server_environment: -s servername to resolve server names different from host where logged on
- cmd/new edrc.no_shell: pseudo shell to prevent shell exits if 'shell' is denied in DENY_LIST of edrc.cfg
- cmd/chg edrc: new command removemenu to remove an existing menu and removescript to remove an existing recovery script. No it is not any more needed to exit to the shell to remove a script or menu
- cmd/chg edrc: \$TODAY is exported to the contributed commands and the recovery scripts, too.
- cmd/chg edrcenv: print the content of the \$TODAY variable, too
- opt/new input: ALPHA, DIGIT, ALNUM, ALNUM+SPACE, ALNUM+_ to allow certain input only
- cmd/chg input: prevent return value to be resolved, as happened when * has been entered, depending on cwd the filenames were returned
- cmd/chg edrc: use option ALNUM+_ in input calls in functions to commands newmenu, newscrip, removescript, removemenu to control inputs
- man/chg edrc.1m: description of .sav directory
- man/chg edrcintro.1: description of .sav directory
- cmd/new stat: return stat() information of a file
- cmd/new bwcreate: create a BarbedWire file to track file changes
- cfg/new bwcreate.cfg: cfgfile for bwcreate
- opt/chg omniutil: -a omnitrig, was not implemented and cannot be realized the way intended to
- cmd/new pack: replaces edrcpack to pack the application into a package
- cmd/chg edrcpack: displays a depreciated message and uses a subset of options of 'pack'
- cmd/chg scriptextract: scriptextract -> edrc.scriptextract
- man/chg scriptextract.cfg: man page removed
- cmd/chg edrccheck: removed
- cmd/chg edrc: the help command is no longer influenced by the more_file permission
- man/new swvi.1: man page for swvi
- cmd/cor swvi: usage message corrected
- man/chg duvi.1: cross reference to swvi added
- man/chg edrcintro.1: term_cap short description removed
- cmd/chg llcomp: complete redesign, formats bzip2, zip, compress added
- cmd/chg lscomp: complete redesign, formats bzip2, zip, compress added
- cmd/chg catcomp: complete redesign, formats bzip2, zip, compress added
- man/chg llcomp.1: adjusted to enhanced functionality
- man/chg lscomp.1: adjusted to enhanced functionality
- man/chg catcomp.1: adjusted to enhanced functionality
- man/chg EDRC.1: zip, unzip added
- man/chg edrcintro.1: zip, unzip added
- cmd/chg is_running: sbit set to allow non-root users to resolve if the package is running. The filepermissions are checked by is_running
- cmd/chg busy: now multiple os aware.

1.5.00

Sun Jul 10 19:45:42 METDST 2005 release

1.4.08

Sun May 22 17:34:30 CEST 2005 release

- cmd/chg edrc: changes bold settings in prompt on terminal changes
- cmd/chg term_cap: removed
- cmd/new lscp: similar to lsmv
- cmd/chg filedist, rcmd: is now able to resolve hostgroups as: @GROUPNAME on interactive input
- cmd/chg pkg_hostname: set AWK to support the SunOS exception
- cmd/chg dbrep, job, joblog, jobwatch, pkg_hostname, ini.*, mkuser : use hostname instead of uname -n
- cmd/chg sysconfig: elimination of // in file output, output of write settings step
- cmd/chg edrcpack: can now create shell archives, too
- cfg/chg kshrc: cmc, cddoc alias
- cmd/chg rcmd: avoid local wildcard resolution on interactive command input, if configured: set path on remote hosts equal to local, invoke commands on remote hosts as script, execution shell is now configurable.
- cfg/new rcmd.cfg: options SET_PATH and EXEC_SHELL.
- cmd/chg edrcpack: shell archive enhanced, on -a only create .sh archive
- cmd/chg shell archive: proper cleanup on termination
- cmd/chg passwdsyncd: .tar.gz -> .cpio.gz
- cmd/chg passwdcyncd_apply: .tar.gz -> .cpio.gz
- doc/chg install-steps: complete review
- cmd/chg passwdsyncd, passwdsyncd_apply: test on HP-11(i) and review
- cmd/chg name: follow symlinks
- cmd/chg revision: follow symlinks
- cmd/new whereami: display server environment and user information of the system where logged on
- cmd/new edrc/contributed: whereami
- man/chg edrcintro: whereami added
- cmd/new trash, untrash: to trash and restore trashed files
- cmd/fix watchdog: list running watchdogs
- opt/chg watchdog: -r interval to control logging or check state
- cmd/new cltrash: clean trash (purge trash / remove trash info files)
- cmd/chg edrcpack: timestamp file format changed. Old format is still recognized to read. There is no need to edit old timestamps.
- cmd/chg scriptextract: adjust timestamp after script installation
- cmd/new syscp: copy a system file to the sysconfig repository
- cmd/new sysvi: equals to svi to unify command naming
- cmd/chg logrotate: all users are allowed to start the command
- cmd/chg logtail, logview: command reads also ~/.mylog_files and ~/.log_files
- cmd/new swvi: create/edit the ..sw_inventory file
- cmd/new sw_report: create a software inventory report.
- cmd/chg cfg2html: cd /tmp during command execution to prevent cfg2html to penetrate the system with tempfiles
- cmd/chg cfg2html_<os><version>: current versions of cfg2html for OS HP-11i, Linux, Solaris
- cmd/chg edrc: correct handling of wild cards in input prompt
- cmd/chg whereami: new fields SU_FROM, SU_TO
- cmd/new user_info: command to read user definition information. This command partly replaces functionality of the whereami command.
- cfg/new user_info.cfg: migrated from whereami.cfg
- cmd/chg whereami: user information resolving moved to command user_info
- cfg/chg whereami.cfg: completely replaced by user_info.cfg.
- man/chg edrcintro.1: Sections COMMANDS, FILE, SEE ALSO.
- doc/new edrc_manpages: in HTML format, locally browseable.
- cmd/new apply2sw_inventory: apply a data stream to a software inventory file.
- cmd/new fields2swvi: convert a data stream to a swvi input sequence.

- cmd/new is_writeable: check if write access to a file or directory is possible.
- cmd/new filelink: create hardlinks between files in two different directories.
- cmd/new lsof: list open files.
- cmd/fix edrcpack: exclude core files but do not exclude core directories.
- cmd/chg edrcpack: enhance usage message with more example usages.
- cmd/chg edrc: possibility to list the files to be distributed ahead of the distribution.
- cmd/fix edrc: evaluation if a file is empty.
- cmd/chg sat: session name changed from SYSTEM_ADMINISTRATION__PID-\$\$ to SAT-\$\$
- cmd/chg logrotate: use library log function, therefore log format changed.
- opt/chg makeuser: -U uid
- cmd/new role_option: to support role based user creation from templates
- cfg/new role_option.cfg: config file for role_option
- cfg/new edrc.cfg: new options RECOVERYPOINTINTIME_MESSAGE, SETENVIRONMENT_MESSAGE
- man/chg edrc.cfg.4: options RECOVERYPOINTINTIME_MESSAGE, SETENVIRONMENT_MESSAGE
- cmd/new homedir: evaluates the homedir of a user
- man/new homedir:
- cmd/chg shar-header, install: log inputs to logfile during installation
- man/new pid:
- man/new woist:
- cmd/new rl: shortcut for rlogin
- cmd/fix catcomp: use GNU cpio even if started outside of edrc

1.4.07

Sun Nov 7 11:16:36 MET 2004 release

- cmd/chg almost all: support a bigger range of operating system ids of one operating system, as Solaris -> Solaris*, Linux -> *Linux*, HP-11|HP-11i -> HP-11|HP-11i|HP-*
- cmd/chg HP-11i/cpio: replace HP11i/cpio with version 2.5
- man/new HP-11i/man1/cpio.1: manpage included
- cmd/chg lscomp, llcomp, catcomp: can now handle .tar.gz and .cpio.gz archives
- cmd/chg edrc: uses now .cpio.gz archive format to distribute scripts
- man/chg edrc: distribution file tar.gz -> cpio.gz
- man/chg edrcintro: enhancement of lscomp, llcomp, catcomp
- cmd/chg sysconfig: backup file tar.gz -> cpio.gz
- cmd/chg passwdcombine: passw save file tar.gz -> cpio.gz
- opt/chg hostlist: remove -s option, add -g option to specify a hostgroup
- cfg/chg hostlist.cfg: HOSTGRPS="@GRPA: host1 host2 @GRPX: host6 host7"
- man/cor sysconfig: typing errors
- cmd/chg edrcpack: edrcpack timestamps in var/edrcpack/ts-* are excluded from generated patch packages
- opt/new edrcpack: -e to exclude scripts directory from a patch archive
- cmd/chg sysconfig: allow "_" in configuration name and configuration description
- cmd/fix sysconfig: correct handling of multiple configuration display

1.4.06

Sat Jul 24 18:16:32 / Sun Aug 8 18:25:47 CEST 2004 release

- opt/new sysconfig: -o OSID, to apply a configuration to the system which is different from the result of the osid command.
- opt/new edrcpack: -p version, to create patch-packages
- cmd/fix fssum: can now handle multiple line bdf output
- cmd/new term_cap: return a termcap value
- cmd/new cdetc, cdvar: cd to application etc and var directory in shell
- cmd/new mkuser, makeuser: integrated into EDRC
- cmd/new server_environment: to print the environment a server belongs to
- man/chg edrcintro: add short descriptions for new commands, add edrcpack directory description in FILES section

- cmd/new xlog: write stdin to a log
- cmd/new pkg_hostname: return packagename or hostname depending on pwd
- cmd/new omniutil: a collection of OBII enhancements
- cmd/new logtail: tail -f common logfiles
- cmd/new logview: view common logfiles
- cmd/new watchdog: a watchdog
- cmd/new passwdsyncd: synchronize passwords

1.4.05

- undated release
- cmd/fix sysconfig: fix stty: this is no tty error.
- man/cor cronhandler: description of action=... files.

1.4.04

- undated release
- cmd/chg edrc: OpenSSH test, internal enhancements
- man/chg edrc: securtiy/<DIST_USER>/ description
- cmd/chg edrcpack: writes now to a logfile
- man/fix sysconfig: logfile description
- cmd/chg filedist: internal enhancements
- cmd/chg rcmd: OpenSS test, internal enhancements

1.4.03

- undated release
- cmd/fix sysconfig: config_byhand bug
- cmd/fix is_config_byhand: proper cleanup of tempfiles
- cfg/fix kshrc: export of HOST

1.4.02

- undated release
- cmd/chg sysconfig: can now handle config_byhand functionality
- opt/new sysconfig: -f option to force configuration (=ignore config_byhands)
- cmd/new rel2abs: convert a relative to an absolute filename
- cmd/new is_config_byhand: evaluate if a file is protected by a config_byhand file
- man/cor cronhandler: FILE section
- man/chg EDRC, edrcintro: new commands added
- man/chg sysconfig: config_byhand description
- cmd/fix bin/* lib/*: permissions and groups corrected

1.4.01

- undated release
- cmd/chg cronhandler: option -a write to save cron/at without removal from system
- man/chg cronhandler: according to the change

1.4.00

- Mon Apr 5 18:44:53 METDST 2004 release
- cmd/chg: replacement of edrcroot thru approot in all commands
- cmd/chg: removal of alpha command pwsync
- cmd/new tty_variable: to query tty settings
- cmd/chg tty_columns: internal use of tty_variable

1.3.08

- Fri Mar 26 10:46:59 MET 2004 release
- cmd/new approot: identical to edrcroot, which will be replaced completely soon.
- man/cor: permission correction of all manpages

1.3.07

- Fri Feb 27 12:43:30 MET 2004 release
- man/cor cronhandler: files corrected
- cmd/fix envpasswdstrip: rootdir evaluation corrected

1.3.06

Tue Feb 17 00:16:49 MET 2004 release

- cmd/new hostlist: return a list of hosts. Used to generalize configuration files
- cmd/new ypxfr_all, passwdsort: taken from tools-eds

1.3.05

undated release

- cmd/fix sysconfig: write correct settings file header

1.3.04

undated release

- cmd/new pkzip: for Linux, Solaris
- cmd/chg: uat and yauat related files moved to doc/notes/dvlp due to incomplete development state of those commands

1.3.03

undated release

- cmd/new pkzip: for HP-11, HP-11i

1.3.02

Fri Jan 30 17:58:12 MET 2004 release

- cmd/chg edrc: version and patchlevel moved to etc/edrcrevision.cfg
- cfg/new edrcrevision.cfg: new configuration file holding the current version and patchlevel of WA2L/edrc
- cmd/chg fssum: cluster package checks will be executed only if the underlying operating system supports it. Therefore rows with the 'p' option in the fssum.cfg file are ignored on systems which do not support cluster packages or for which the PACKAGE_CHECK function is not implemented within fssum.

1.3.01

Fri Jan 9 15:22:33 MET 2004 release

- cmd/chg passwdcombine: can handle now trusted system and /etc/passwd systems and is able to do a more filigree selection.
- cfg/new edrc: NOTRECOGNIZED_MESSAGE option to generate a WARNING message if a user entered a command in edrc that does not exist.

1.3.00

Sun Nov 2 12:17:50 CET 2003 release

- cmd/new edrc, filedist, rcmd: prepared for secure shell
- cfg/new edrc.cfg, filedist.cfg: DIST_MODE to configure rtools or OpenSSH
- cfg/new rcmd.cfg: EXEC_MODE to configure rtools or OpenSSH

1.2.01

undated release

- opt/new edrcpack: -s = split package into peaces which fit to a diskette

1.2.00

Sun Oct 26 15:55:14 CET 2003 release

- cmd/new fit2width: fit a stream to a width
- cmd/new tty_columns: evaluate tty width
- cmd/fix days, input, choice: path to osid command

1.1.04

undated release

- cmd/new fssum: generate filesystem summaries
- cmd/new edrccheck: check needed connections and permissions for edrc
- cmd/new days: evaluate days since Jan 1st
- cmd/chg is_running: is able to resolve aliases now, some path settings added
- man/cor sysconfig: Optional: NO in FILES Section, <OSID>/root

1.1.03

Sat Oct 11 15:25:17 MEST 2003 release

- new: HP-11i support. HP-11i will probably cover many of future HP UNIX releases.

1.1.02

Mon Sep 29 16:30:28 METDST 2003 release

- cmd/new dbrep: as enhancement of the no longer maintained dbstat command
- cmd/new ini, ini.<shell>: ad hoc environment initialization

1.1.01

Tue Sep 16 14:22:18 MEST 2003 release

- cmd/chg oerr: new option TNS

1.1.00

Tue Sep 2 16:46:56 MEST 2003 release

- cmd/fix cronhandler: exit if no user dirs in homedir

1.0.21

Mon Aug 18 09:12:20 METDST 2003 tmp release

- cmd/chg edrcman: supports OSID dependent manpages now

1.0.20

Wed Aug 6 15:07:53 MEST 2003 release

- cmd/new edrc: title to modify a menu title
- cmd/new tcpdump integrated
- opt/new filedist: can now block (prevent) files from distribution to/from certain hosts
- cmd/new oerr contributed command
- cmd/chg proc contributed command renamed to pid
- doc/chg install-steps: enhanced
- man/new osid (3)
- man/new osid.dat (4)

1.0.18

Sat May 10 19:49:29 CEST 2003 release

- cmd/fix edrc: syntax error correction in function set_terminal()

1.0.16

Mon Apr 14 13:49:47 METDST 2003 release

- cmd/new edrc: @ command which invokes the last menupoint chosen
- cmd/new edrc: EDRC_NLS_DATE_FORMAT, EDRC_NLS_LANG is exported to contrib commands, scripts and shell to avoid variable overwriting by recovery scripts
- cmd/new edrc: EDRC_DIST_USER is exported to contrib commands, scripts and shell. This variable is equal to DIST_USER in the configuration file cmd/new job, joblog, jobwatch: integrated into EDRC

1.0.15

Wed Apr 2 16:17:31 METDST 2003 release

- opt/new edrc: -t option to skip terminal initialization
- cmd/fix edrc: test of start user relocated in program, log output to syslog if edrc is invoked with the wrong user
- cmd/new lsmv: print a list of mv-commands to screen

1.0.14

Mon Mar 24 19:52:46 MET 2003 release

- cmd/chg edrc: prompt includes the hostname
- cmd/new edrc: erase command to set the erase terminal character

1.0.12

Tue Mar 18 21:07:15 MET 2003 release

- cmd/fix .os_wrapper: more accurate command evaluation
- cfg/chg kshrc: terminal settings
- cmd/chg edrc: terminal settings to ensure proper backspace handling
- cfg/chg edrc.cfg: TERM setting is now the default if the terminal cannot be evaluated by the system.
- cmd/chg edrc: xterm is temporarily substituted by tee -a to save the logs due to SIGSEGV error on HP-UX 11.00
- cmd/chg sysconfig: does handle the check file now (again)

- cfg/new kshrc: variable \$p in shell is the last visited directory ahead of exiting the shell. This variable is displayed if 'abc' is invoked in the shell.
- cfg/new kshrc: suspend signal is now handled in shell (^Z).
- man/cor catcomp.1, edrc.1m, edrc.cfg.4

1.0.09

Sat Mar 8 02:25:16 CET 2003 release

- cmd/new filedist: copy files to other hosts
- cmd/chg edrc: distribution hostlist has a leading and trailing space
- cmd/new logcut: shorten logfiles
- cmd/new locations: list corporate locations with local time
- cmd/new is_running: check if a cluster package is up

1.0.08

Thu Mar 6 14:24:04 MET 2003 release

- opt/new edrcpack: -t
- several bugfixes and justifications

1.0.07

Wed Mar 5 20:50:31 MET 2003 release

- cmd/fix: recursive login bug existing on HP-UX

1.0.00

Tue Mar 4 00:13:16 CET 2003 release

- chg/all owner changed to root:sys
- 1st semi-productive release

0.9.35

Sun Mar 2 21:09:49 CET 2003 release

- cmd/new svi: edit files in a sysconfig repository
- cmd/chg scriptextract: configfile is relative to edrc/etc now
- cmd/chg edrc: configfile is relative to edrc/etc now
- cmd/chg edrc: configfile is submitted to scriptextract while distributing scripts using the "distribute" EDRC command
- cmd/cor sysconfig: several message changes
- cmd/new sat: start edrc with a configuration for system administration
- man/cor man pages existing in 0.9.11 according to the correction hints of Craiq Schmid
- cmd/new contributed/countdown: a countdown handler
- cmd/new edrc: new directory var/contrib/edrc
- cmd/new edrc: EDRC_CONTRIB_VARDIR is exported to contributed commands

0.9.31

undated release

- cmd/new edrc: export of EDRC_SCRIPTS_BASEDIR to contrib commands, shell and recovery scripts.
- cmd/new edrcenv: print of EDRC_SCRIPTS_BASEDIR
- opt/new edrc: -s option which allows silent startup

0.9.29

undated release

- cfg/new sysconfig: backup filelist
- cmd/new xtee: is now a compiled c program due to troubles with the awk version.

0.9.22

undated release

- cmd/new edrc: check of SCRIPTS_BASEDIR, exit 7 if fail
- opt/new edrc: -n session_name

0.9.18

Fri Feb 14 02:11:42 CET 2003 release

- cmd/new edrc: history
- cmd/new edrc: busy, !!, !<num>
- cfg/new edrc.cfg: SHELL_LENGTH, SHELL_SEARCH

- cmd/fix edrc: signal handling ported to UNIX

0.9.11

Tue Feb 11 22:10:25 MET 2003 release

- man/cor
- man/new catcomp, edrc.help, lscomp, llcomp, edrcroot
- cmd/fix edrc: contrib commands log to the main logfile edrc.log

0.9.10

Sun Feb 9 14:00:11 CET 2003 release

- cmd/new edrc: ABORT_MODE, kill with USR1 and USR2
- man/chg edrc.lm: completed

0.9.09

Sat Feb 8 21:46:12 MET 2003 release

- cmd/chg edrc: the current PID is session name if no name is set.
- cmd/new edrc: log header output is configurable thru LOGHEAD_OUTPUT.
- opt/chg scriptextract: -v to -V for version output
- man/new scriptextract.cfg
- cmd/chg patchlevel numbers changed to two digits
- man/cor typing errors

0.8.1 ... 0.9.8

Other development versions.

0.8.0

Sat Feb 1 23:20:28 MET 2003 release

0.7.0

Sat Feb 1 23:20:28 MET 2003 release

0.2.0 ... 0.6.0

Other development versions.

0.1.0

First development version.

SEE ALSO

edrcintro(1), **EDRC(1)**

NOTES

-

AUTHOR

WA2L/edrc was developed by Christian Walther. Send suggestions and bug reports to wa2l@users.sourceforge.net .

COPYRIGHT

Copyright © 2024 Christian Walther

This is free software; see **edrc/doc/COPYING** for copying conditions. There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.